



NRL/MR/6040--17-9765

A Moving Discontinuous Galerkin Finite Element Method for Flows with Interfaces

ANDREW CORRIGAN

ANDREW D. KERCHER

DAVID A. KESSLER

*Laboratories for Computational Physics and Fluid Dynamics
Materials Science and Component Technology Directorate*

December 7, 2017

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE (DD-MM-YYYY) 07-12-2017	2. REPORT TYPE Memorandum Report	3. DATES COVERED (From - To)		
4. TITLE AND SUBTITLE A Moving Discontinuous Galerkin Finite Element Method for Flows with Interfaces		5a. CONTRACT NUMBER		
		5b. GRANT NUMBER 64-4493-07		
		5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S) Andrew Corrigan, Andrew D. Kercher, and David A. Kessler		5d. PROJECT NUMBER		
		5e. TASK NUMBER		
		5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Research Laboratory 4555 Overlook Avenue, SW Washington, DC 20375-5320		8. PERFORMING ORGANIZATION REPORT NUMBER NRL/MR/6040--17-9765		
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Naval Research Laboratory 4555 Overlook Avenue, SW Washington, DC 20375-5320		10. SPONSOR / MONITOR'S ACRONYM(S) NRL		
		11. SPONSOR / MONITOR'S REPORT NUMBER(S)		
12. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.				
13. SUPPLEMENTARY NOTES Submitted for consideration for publication in the Journal of Computational Physics on October 26, 2017.				
14. ABSTRACT A moving discontinuous Galerkin finite element method with interface condition enforcement (MDG-ICE) is formulated for flows with discontinuous interfaces. The underlying weak formulation enforces the interface condition separately from the conservation law, so that the residual only vanishes upon satisfaction of both, while treating the discrete grid geometry as a variable. In contrast to the standard discontinuous Galerkin (DG) method, this method has both the means to detect, via interface condition enforcement, and satisfy, via grid movement, the conservation law and its associated interface condition. The method therefore directly fits interfaces, including shocks, preserving a high-order representation up to the interface, without requiring shock capturing or an upwind numerical flux to achieve stability. It can be generalized to flows with a-priori unknown interfaces with non-trivial topology and curved interface geometry as well as to an arbitrary number of spatial dimensions. Unsteady flows are represented in a manner similar to steady flows using a spacetime formulation. In addition to computing flows with interfaces, the method can represent point singularities in a flow field by degenerating cuboid elements. In general, the method works in conjunction with standard local grid operations, including edge collapse, to ensure that degenerate cells are removed. Test cases are presented for up to three-dimensional flows that provide an initial assessment of the stability and accuracy of the method.				
15. SUBJECT TERMS High order finite elements; Discontinuous Galerkin method; Interface condition enforcement; Grid adaptivity; Shock-fitting; Spacetime				
16. SECURITY CLASSIFICATION OF: a. REPORT Unclassified Unlimited		17. LIMITATION OF ABSTRACT b. ABSTRACT Unclassified Unlimited	18. NUMBER OF PAGES c. THIS PAGE Unclassified Unlimited	19a. NAME OF RESPONSIBLE PERSON Andrew Corrigan
				19b. TELEPHONE NUMBER (include area code) (202) 767-3640

CONTENTS

EXECUTIVE SUMMARY	E-1
1. INTRODUCTION	1
1.1 Background	1
2. STANDARD DISCONTINUOUS GALERKIN METHOD FOR FLOWS WITH INTERFACES	3
3. MOVING DISCONTINUOUS GALERKIN METHOD WITH INTERFACE CONDITION ENFORCEMENT	6
3.1 Formulation in physical space with fixed geometry	6
3.2 Formulation in reference space with variable geometry	9
4. SOLVER	12
4.1 Discretization	12
4.2 Nonlinear solver	13
4.3 Grid Operations	14
4.4 Entropy Condition Enforcement	14
5. EXAMPLES	15
5.1 Moving Interface	15
5.2 Spacetime Burgers flow	16
5.3 Shock tube	18
5.4 Intersecting oblique shocks	20
5.5 Bow Shock	22
6. CONCLUSIONS AND FUTURE WORK	27
REFERENCES	31

FIGURES

1	A moving interface with spatial velocity $v_x = \frac{1}{10}$	4
2	A moving interface with spatial velocity $v_x = \frac{1}{10}$	17
3	A moving interface with spatial velocity $v_x = \frac{3}{2}$	18
4	Burgers spacetime shock formation.....	19
5	Sod shock tube	21
6	Intersecting oblique shocks: Grid	23
7	Intersecting oblique shocks: Density	24
8	Intersecting Oblique Shocks: Entropy.....	25
9	Three-Dimensional Intersecting Shocks	26
10	Bow Shock: Grids.....	28
11	Bow Shock: Pressure Field	29
12	Bow Shock: Stagnation Enthalpy Field.....	30
13	Bow Shock: Line Plots.....	31

TABLES

1	Intersecting Oblique Shocks	20
---	-----------------------------------	----

EXECUTIVE SUMMARY

A moving discontinuous Galerkin finite element method with interface condition enforcement is formulated for flows with discontinuous interfaces. The method preserves a high-order representation up to the interface, without relying on artificial dissipation or shock capturing for stabilization. The method can compute curved interfaces, with a priori unknown topology, in multiple dimensions. The method applies to both steady flows, or unsteady flows via a spacetime formulation.

A MOVING DISCONTINUOUS GALERKIN FINITE ELEMENT METHOD FOR FLOWS WITH INTERFACES

1. INTRODUCTION

High-order methods have the potential to resolve complex flow field structures with high accuracy and at a reduced computational cost compared to lower-order methods. One such method in particular, the discontinuous Galerkin finite element method (DG) [1–9], has been used with increasing success in the field of computational fluid dynamics over the past two decades to simulate a wide variety of flows due, in no small part, to its ability to achieve high-order accuracy on fully unstructured grids. Another advantage of the DG method is a well-developed theory of adjoint consistency [9–12], which makes it a powerful tool for adjoint-based optimization. In addition, spacetime discontinuous Galerkin methods [13–15] provide a unified discretization of unsteady flow problems by simultaneously discretizing space and time. Thus, spacetime DG offers the prospect of both high-order accuracy in space and time *and* adjoint consistency, which provides an ideal framework for the optimization of unsteady flows. One question that naturally arises is how these remarkable properties fare in inviscid flows that are not smooth and contain discontinuous interfaces, including material interfaces and shocks. Since DG employs a discrete function space composed of discontinuous, piecewise polynomials, it can, in principle, directly represent these discontinuous interfaces. However, this requires that the interfaces are aligned with the grid. When misaligned, instabilities can arise, leading to the common approach of artificial stabilization or shock capturing, cf. [16–19]. While shock capturing has been remarkably successful for simulating a variety of complex flows, it can diminish some of the benefits of DG, including high-order accuracy and consistency, an ongoing issue emphasized in a recent survey of higher-order methods by Wang et al. [20].

In order to extend the advantages of DG from smooth to piecewise smooth flows with a priori unknown interfaces, we propose a moving discontinuous Galerkin finite element method with interface condition enforcement (MDG-ICE). This method fits, rather than captures interfaces, so that properties such as high-order accuracy and adjoint consistency might be preserved. There are two important features of MDG-ICE that distinguish it from standard DG. First, the method enforces the interface condition separately from the conservation law in the underlying weak formulation, so that the residual only vanishes upon satisfaction of both. Second, the method treats the discrete grid geometry as an additional solver variable so that the solver simultaneously moves the discrete grid geometry to fit interfaces while resolving the flow field. Thus, in contrast to standard DG, this method has both the means to detect, via interface condition enforcement, and satisfy, via grid movement, the conservation law and its associated interface condition.

1.1 Background

Interface, or more specifically, shock fitting is a technique with a long history within the field of computational fluid dynamics, as surveyed by Moretti [21] and Salas [22, 23]. While a large number of variations exist, the basic idea of these methods is to explicitly incorporate the definition of a shock surface into the underlying grid structure and then move the grid according to the shock speed dictated by the Rankine-Hugoniot conditions. Recent work by Paciorri, Bonfiglioli, et al. [24–28] has greatly extended the range of

application of these methods and clearly demonstrated the advantage of preserving design accuracy in the vicinity of a shock, a property sacrificed by standard shock-capturing methods. An ongoing difficulty with existing shock-fitting approaches is treating shocks or interfaces whose topologies are unknown *a priori* or whose topologies change in time.

Simulating material discontinuities poses an even more significant computational challenge. Such interfaces, which lack the nonlinear self-steepening behavior of shock waves, are highly susceptible to smearing caused by artificial dissipation in shock-capturing methods. In an attempt to avoid this issue, a number of techniques have been developed that instead track the motion of the interface explicitly. Front tracking methods, introduced by Glimm et al. [29, 30] and extended to DG by Nguyen et al. [31], analogously to shock fitting, are particularly well-suited for tracking the evolution of material or contact discontinuities. Rather than requiring interfaces to be aligned with grid interfaces, the extended finite element method (XFEM) [32] uses local enrichment functions to represent interfaces that require special quadrature rules to perform the numerical integration. A broader family of unfitted finite element methods (FEM) [15, 33–38] have also been developed for moving interface problems. An advantage of these methods is that by avoiding the alignment of grid interfaces with flow interfaces, it becomes less onerous to simulate flows with unknown or dynamic interface topology, while providing high-order accuracy, especially in comparison to interface capturing approaches.

Lagrangian or arbitrary Lagrangian-Eulerian (ALE) methods, represented for higher-order finite element methods by work such as that of Dobrev et al. [39–41], Persson et al. [42], and Fidkowski [43], are another class of interface-fitting techniques. These methods have numerous advantages, particularly regarding highly accurate and robust treatment of dynamic material interfaces or moving bodies. Lagrangian methods excel at representing such material interfaces directly, as well as moving bodies, since they are based on moving a spatial grid with the velocity of the flow. ALE methods further improve upon Lagrangian methods, by allowing for more flexibility in the grid velocity. However, in the presence of shocks, some form of shock capturing appears to still be required. In addition, for unsteady flow problems, ALE methods can deform spatial grids so severely that, between time steps, a possible re-interpolation operation is required to preserve validity of the grid. In addition to possibly introducing error, such a re-interpolation process also complicates adjoint analysis. In contrast, within a spacetime formulation both stationary and spatially deforming boundaries are readily treated as static spacetime boundaries, avoiding the difficulties of ALE for highly deforming geometry [44], while inheriting the adjoint analysis of DG that has been developed for steady flows.

Other related methods are those that also involve moving the grid geometry to improve accuracy. Harten and Hyman [45] solve one-dimensional unsteady conservation laws using a moving grid based on local characteristic velocities. The original moving finite element method introduces [46–48] the idea of treating a spatial grid as a solver variable, which is integrated in time for unsteady flows. Roe and Nishikawa also proposed a fluctuating splitting method which involves treating the grid as a variable [49]. Recently, Sanjaya and Fidkowski [50] proposed an element warping strategy, which like the present work also moves element nodes as part of the solver, greatly improving solution accuracy, as guided by least-squares or output-based error metrics.

While the aforementioned interface-fitting techniques have proven useful for a number of flow scenarios, a number of difficulties still exist that limit their general applicability. In particular, solving unsteady flows with complex, dynamic interfaces that undergo topological changes remains a challenge. The MDG-ICE

method attempts to circumvent these difficulties by making interface fitting an intrinsic part of the underlying formulation in which the flow field and interface locations are solved as part of a single unified method. By utilizing a spacetime formulation, complex interface dynamics that can cause extreme grid distortion using standard time-marching interface-fitting techniques can be handled by making relatively small modifications to an existing spacetime mesh. Furthermore, incorporating shock fitting directly into a unified variational finite element formulation would provide a natural setting to formulate adjoint-based optimization in the presence of shocks [51–54], and enable differentiability with respect to interface geometry. Point singularities that arise in inviscid flows (e.g., rarefactions and shock formations in spacetime) can also trigger instabilities, due to the inability of standard polynomial basis functions to represent the instantaneous transition that occurs at a single point. While such singularities can be stabilized using the same artificial dissipation introduced by shock capturing, they provide a source for artificial entropy generation which contributes to the reduction in the order of accuracy of the underlying numerical method. MDG-ICE also provides a means of naturally treating these types of singularities. Boundary layers, which arise in viscous flows, also require significant computational effort to resolve accurately and stably. Therefore, a broader goal beyond a method tailor-designed for shocks, would be a general method that can exploit grid movement to help resolve a range of challenging flow features including interfaces, singularities, and boundary layers.

2. STANDARD DISCONTINUOUS GALERKIN METHOD FOR FLOWS WITH INTERFACES

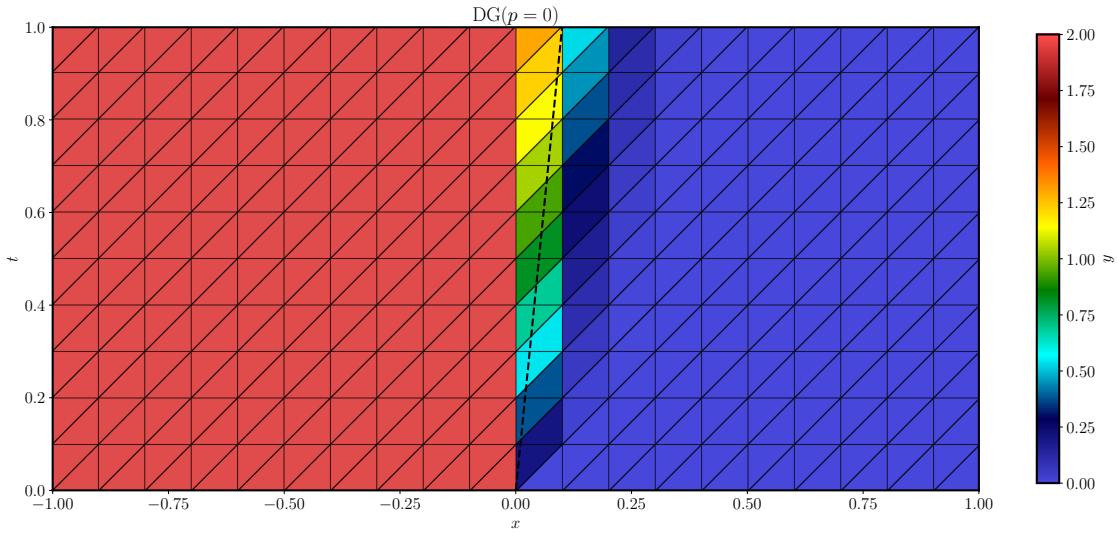
The discontinuous Galerkin method, which is based on piecewise polynomial approximation over potentially unstructured grids, is an ideal candidate for computing piecewise smooth flows with interfaces. As long as any interfaces present in the exact flow solution coincide with the interfaces present in the grid, DG has the potential to approximate such flows with high order accuracy. However, we seek a method that not only has the means to represent piecewise smooth flows, but also provides the means to detect interfaces. Therefore, in order to compute flows with a priori unknown interfaces, we seek a method for which the residual vanishes only upon satisfaction of the interface condition, or a weak or variational equivalent thereof. With such a method at hand, the solver can then introduce the discrete grid geometry as a variable, so that as the residual is driven to zero, the interfaces are fit with increasing accuracy.

To demonstrate the importance of the residual vanishing only upon satisfaction of the interface condition, we consider a linear advection problem for flow containing a jump discontinuity in a two-dimensional domain, with an imposed uniform velocity of $v = (v_x, v_t) \in \mathbb{R}^2$. We assume that $v_t = 1$ and interpret the second dimension as representing time, while we assume that the spatial velocity is positive, $v_x > 0$. Therefore, we can also interpret the interface as an unsteady, moving interface in one-dimensional space. Since the flow is piecewise smooth, we decompose the domain Ω into two subdomains Ω_1, Ω_2 , separated by an interface $\partial\Omega_1 \cap \partial\Omega_2$ with a normal $n = (n_x, n_t) \in \mathbb{R}^2$ that is perpendicular to the spacetime velocity (v_x, v_t) . The two subdomains are defined as

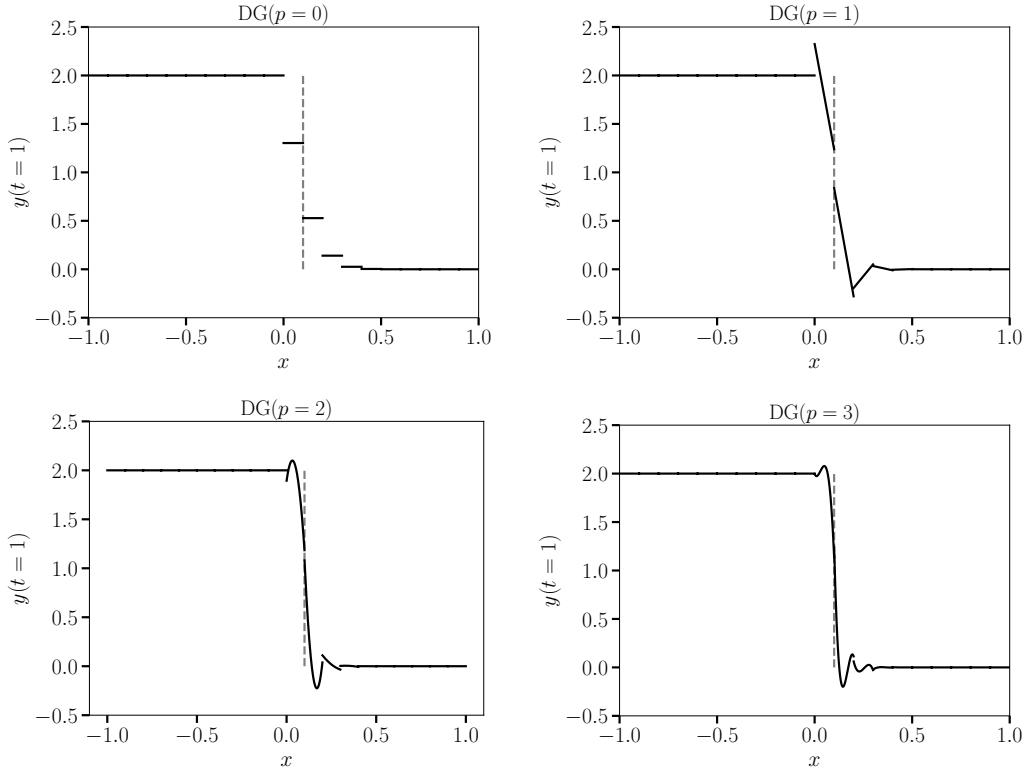
$$\begin{aligned}\Omega_1 &= \{(x, t) \in \Omega \mid (x, t) \cdot (n_x, n_t) < 0\} \\ \Omega_2 &= \{(x, t) \in \Omega \mid (x, t) \cdot (n_x, n_t) > 0\}.\end{aligned}\tag{1}$$

The governing conservation law

$$\nabla \cdot \mathcal{F}(y) = \nabla \cdot (v_x y, v_t y) = 0, \text{ in } \Omega_1, \Omega_2\tag{2}$$



(a) The spacetime solution obtained using $\text{DG}(p = 0)$. The interface present in the exact solution is illustrated with a dashed line.



(b) The piecewise polynomial solutions obtained for $\text{DG}(p = 0, 1, 2, 3)$, sampled along the line $t = 1$. The dashed line located at $x = \frac{1}{10}$ spans the interface present in the exact solution.

Fig. 1: A moving interface with spatial velocity $v_x = \frac{1}{10}$. We observe that the residual has vanished without having fit the interface.

is augmented with the initial condition

$$y(x, t = 0) = \begin{cases} 2 & x < 0 \\ 0 & x > 0 \end{cases}, \quad (3)$$

inflow condition

$$y(x = -1, t) = 2, \quad (4)$$

as well as the interface condition

$$[n \cdot \mathcal{F}(y)] = 0 \text{ on } \partial\Omega_1 \cap \partial\Omega_2. \quad (5)$$

The exact solution, satisfying Eqs. (2), (3), (4), (5), is the piecewise constant flow

$$y(x, t) = \begin{cases} 2 & \text{in } (x, t) \in \Omega_1 \\ 0 & \text{in } (x, t) \in \Omega_2 \end{cases}. \quad (6)$$

We now evaluate the criterion that the residual vanishes only upon satisfaction of the interface condition, for the case of the standard discontinuous Galerkin method where the grid is misaligned with the interface present in the exact solution. Let \mathcal{T} be a partition of Ω , consisting of disjoint cells κ , so that $\overline{\Omega} = \cup_{\kappa \in \mathcal{T}} \overline{\kappa}$. Also let Y, V be piecewise polynomial function spaces defined over \mathcal{T} , cf. Section 4.1. The standard DG weak formulation is given: find $y \in Y$ such that

$$\sum_{\kappa \in \mathcal{T}} ((h(y^+, y^-, n), v^+)_{\partial\kappa} - (\mathcal{F}(y), \nabla v)_{\kappa}) = 0 \quad \forall v \in V, \quad (7)$$

where the standard upwind numerical flux for linear advection is

$$h(y^+, y^-, n) = \begin{cases} (n \cdot v) y^+ & \text{if } n \cdot v < 0 \\ (n \cdot v) y^- & \text{if } n \cdot v \geq 0 \end{cases}, \quad (8)$$

which is also specialized at the domain boundary $\partial\Omega$ in order to incorporate boundary conditions. For a detailed formulation and analysis of this method, for the case of smooth flow, see Hartmann and Leicht [9, 11] and the references contained therein. Figure 1 depicts the solution computed using DG for the case of $v_x = \frac{1}{10}$, along with the intentionally misaligned grid and interface present in the exact solution, $\partial\Omega_1 \cap \partial\Omega_2$. Despite this misalignment, the residual vanishes to machine precision. We observe that, in the lower-order case, the solver produces a monotonic solution, yet the interface is diffused over many cells, which is representative of the stability of low-order methods even in the presence of shocks. In the higher-order case, the solution is less diffused, however, over-shoots and under-shoots are present, which is indicative of the instability of high-order methods in the presence of discontinuous interfaces, necessitating shock capturing strategies that locally stabilize the solution while, at best locally, sacrificing high-order accuracy [20].

For both high and low order approximations, using the standard DG formulation, the residual vanishes without having fit the interface. We therefore conclude that the standard DG formulation lacks the means to unambiguously detect a priori unknown interfaces thus motivating the pursuit of a revised formulation.

3. MOVING DISCONTINUOUS GALERKIN METHOD WITH INTERFACE CONDITION ENFORCEMENT

We now present a revised formulation of DG, which incorporates the interface condition separately into the discretization in order to satisfy the criterion that the residual only vanishes once the interface condition is satisfied, thus providing a means to detect a priori unknown interfaces. The method, which we refer to as the Moving Discontinuous Galerkin Method with Interface Condition Enforcement (MDG-ICE), is first formulated in physical space in Section 3.1. In order to drive the residual to zero, thus enforcing both the conservation law and its associated interface condition, the method must have a means to fit interfaces, which is provided by allowing the grid to move, or in other words, treating the grid as a variable. Therefore, the formulation is mapped into reference space in Section 3.2, to clearly identify terms with a geometric dependence and facilitate differentiation with respect to geometry.

3.1 Formulation in physical space with fixed geometry

3.1.1 Strong formulation

Let $\Omega \subset \mathbb{R}^n$ be a given domain. We assume that Ω is partitioned by \mathcal{T} , consisting of disjoint sub-domains or cells κ , so that $\bar{\Omega} = \cup_{\kappa \in \mathcal{T}} \bar{\kappa}$, with interfaces ε composing a set \mathcal{E} so that $\cup_{\varepsilon \in \mathcal{E}} \varepsilon = \cup_{\kappa \in \mathcal{T}} \partial \kappa$, over which a normal n is defined. Consider a nonlinear conservation law, given in strong form, defined for piecewise smooth, \mathbb{R}^m -valued, functions y as

$$\nabla \cdot \mathcal{F}(y) = 0 \text{ in } \kappa \quad \forall \kappa \in \mathcal{T}, \quad (9)$$

$$[n \cdot \mathcal{F}(y)] = 0 \text{ on } \varepsilon \quad \forall \varepsilon \in \mathcal{E}, \quad (10)$$

in terms of a given flux function $\mathcal{F} : \mathbb{R}^m \rightarrow \mathbb{R}^{m \times n}$. In general one might consider systems of the form

$$\nabla \cdot \mathcal{F}(y, \nabla y) = \nabla \cdot \left(\mathcal{F}^c(y) - \mathcal{F}^d(y, \nabla y) \right) = \mathcal{S}(y), \quad (11)$$

where the convective flux $\mathcal{F}^c(y)$ is only a function of the state y , the diffusive flux $\mathcal{F}^d(y, \nabla y)$ is also possibly a function of the gradient ∇y , while $\mathcal{S}(y)$ denotes a source term. In the present setting, we restrict the formulation to convection only, so that $\mathcal{F}(y) = \mathcal{F}^c(y)$ and $\mathcal{S}(y) = 0$.

We assume that \mathcal{E} consists of two disjoint subsets: the interior interfaces

$$\varepsilon_0 \in \mathcal{E}_0 = \{ \varepsilon_0 \in \mathcal{E} \mid \varepsilon_0 \cap \partial \Omega = \emptyset \} \quad (12)$$

and exterior interfaces

$$\varepsilon_\partial \in \mathcal{E}_\partial = \{ \varepsilon_\partial \in \mathcal{E} \mid \varepsilon_\partial \subset \partial \Omega \}, \quad (13)$$

so that $\mathcal{E} = \mathcal{E}_0 \cup \mathcal{E}_\partial$. For interior interfaces $\varepsilon_0 \in \mathcal{E}_0$ there exists $\kappa^+, \kappa^- \in \mathcal{T}$ such that $\varepsilon_0 = \partial \kappa^+ \cap \partial \kappa^-$, so that (10) is defined as

$$0 = [\![n \cdot \mathcal{F}(y)]\!] = n^+ \cdot \mathcal{F}(y^+) + n^- \cdot \mathcal{F}(y^-), \text{ on } \varepsilon_0 \quad \forall \varepsilon_0 \in \mathcal{E}_0 \quad (14)$$

where n^+, n^- denote the outward facing normal of κ^+, κ^- respectively, so that $n^+ = -n^-$. For exterior interfaces

$$0 = [\![n \cdot \mathcal{F}(y)]\!] = n^+ \cdot \mathcal{F}(y^+) - n^+ \cdot \mathcal{F}_\partial(y^+), \text{ on } \varepsilon_\partial \quad \varepsilon_\partial \in \mathcal{E}_\partial. \quad (15)$$

Here $n^+ \cdot \mathcal{F}_\partial(y^+)$ is the imposed normal boundary flux, which may or may not be a function of y^+ depending on the type of boundary condition. Therefore, we further decompose \mathcal{E}_∂ into disjoint subsets of inflow and outflow interfaces $\mathcal{E}_\partial = \mathcal{E}_{\text{in}} \cup \mathcal{E}_{\text{out}}$, so that at an outflow interface ε_{out} the boundary flux is defined as the interior flux,

$$n^+ \cdot \mathcal{F}_{\text{out}}(y^+) = n^+ \cdot \mathcal{F}(y^+), \quad (16)$$

and therefore (15) is satisfied trivially, which is equivalent to no boundary condition being enforced. At an inflow boundary $\varepsilon_{\text{in}} \in \mathcal{E}_{\text{in}}$, the normal boundary flux is a fixed constant independent of the interior state y^+

$$n^+ \cdot \mathcal{F}_{\text{in}}(y^+) = n^+ \cdot \mathcal{F}_{\text{in}}. \quad (17)$$

Systems governed by Eqs. (9), (10) include linear scalar advection (2), spacetime Burgers flow (69), and inviscid, compressible (Euler) flow, both in steady and spacetime form. The Euler flow state variable is given by

$$y = (\rho, \rho v_1, \dots, \rho v_n, \rho E) \in \mathbb{R}^m, \quad (18)$$

where $m = n + 2$. The i -th spatial flux component is given by

$$\mathcal{F}_i(y) = (\rho v_i, \rho v_i v_1 + p \delta_{i1}, \dots, \rho v_i v_n + p \delta_{in}, \rho H v_i) \in \mathbb{R}^m, \quad (19)$$

Here ρ is density, $(v_1, \dots, v_n) \in \mathbb{R}^n$ is velocity, ρE is stagnation energy per unit volume, and

$$H = (\rho E + p) / \rho \quad (20)$$

is stagnation enthalpy. The pressure p is defined as

$$p = (\gamma - 1) \left(\rho E - \frac{1}{2} \sum_{i=1}^n \rho v_i v_i \right), \quad (21)$$

where the ratio of specific heats for air is given as $\gamma = 1.4$. The steady Euler flux is

$$\mathcal{F}(y) = (\mathcal{F}_1(y), \dots, \mathcal{F}_n(y)) \in \mathbb{R}^{m \times n}, \quad (22)$$

while the spacetime Euler flux is

$$\mathcal{F}(y) = (\mathcal{F}_1(y), \dots, \mathcal{F}_n(y), y) \in \mathbb{R}^{m \times (n+1)}. \quad (23)$$

3.1.2 Weak formulation

Let the solution space Y be a vector-valued, broken Sobolev space,

$$Y = [H^k(\mathcal{T})]^m = \left\{ y \in [L^2(\Omega)]^m \mid \forall \kappa \in \mathcal{T}, y|_{\kappa} \in [H^k(\kappa)]^m \right\}, \quad (24)$$

where k is a positive integer. The weak formulation is obtained by integrating the conservation law (9) and its interface condition (10) against separate test functions: find $y \in Y$ such that

$$\sum_{\kappa \in \mathcal{T}} (\nabla \cdot \mathcal{F}(y), v)_{\kappa} - \sum_{\varepsilon \in \mathcal{E}} ([n \cdot \mathcal{F}(y)], w)_{\varepsilon} = 0 \quad \forall (v, w) \in V \times W, \quad (25)$$

or, based on an integration by parts, equivalently,

$$\sum_{\kappa \in \mathcal{T}} ((n \cdot \mathcal{F}(y^+), v^+)_{\partial \kappa} - (\mathcal{F}(y), \nabla v)_{\kappa}) - \sum_{\varepsilon \in \mathcal{E}} ([n \cdot \mathcal{F}(y)], w)_{\varepsilon} = 0 \quad \forall (v, w) \in V \times W, \quad (26)$$

where the test space $V = Y$, while W is the corresponding single-valued trace space.

In contrast to the classical discontinuous Galerkin formulation, we do not substitute the flux function in the resulting surface integral over $\partial \kappa$ with a numerical flux, cf. [9, 11], and instead retain the interior flux. Global coupling is instead achieved through the weakly enforced interface condition over each $\varepsilon \in \mathcal{E}$. Since the conservation law and its interface condition are integrated against separate test functions, the residual only vanishes upon weak satisfaction of both. This formulation can also be seen in the analysis of adjoint flow problems involving shocks [51–54].

3.1.3 Alternative weak formulations

Although not considered in this work, the weak formulation (26) could be modified to substitute a numerical flux, while still enforcing the interface condition separately: find $y \in Y$ such that

$$\sum_{\kappa \in \mathcal{T}} ((h(y^+, y^-, n), v^+)_{\partial \kappa} - (\mathcal{F}(y), \nabla v)_{\kappa}) - \sum_{\varepsilon \in \mathcal{E}} ([n \cdot \mathcal{F}(y)], w)_{\varepsilon} = 0 \quad \forall (v, w) \in V \times W, \quad (27)$$

which brings it closer to the standard DG method. In fact, if we choose W to be a zero-dimensional space, then the standard DG formulation (7) is recovered. We observe also that the choice of an interior numerical

flux $h(y^+, y^-, n) = n \cdot \mathcal{F}(y^+)$ recovers (26), while the exterior numerical flux $h(y^+, y^-, n) = n \cdot \mathcal{F}(y^-)$ and integrating by parts, would lead to: find $y \in Y$ such that

$$\sum_{\kappa \in \mathcal{T}} ((\nabla \cdot \mathcal{F}(y), v)_\kappa - ([n \cdot \mathcal{F}(y)], v^+)_{\partial\kappa}) - \sum_{\varepsilon \in \mathcal{E}} ([n \cdot \mathcal{F}(y)], w)_\varepsilon = 0 \quad \forall (v, w) \in V \times W. \quad (28)$$

The proposed formulation is also related to the hybridized DG formulation [55, 56], which introduces an interface variable $\hat{y} \in W$ to obtain a weak formulation: find $(y, \hat{y}) \in Y \times W$,

$$\sum_{\kappa \in \mathcal{T}} ((\nabla \cdot \mathcal{F}(y), v)_\kappa + (h(y^+, \hat{y}, n) - n \cdot \mathcal{F}(y^+), v^+)_{\partial\kappa}) - \sum_{\varepsilon \in \mathcal{E}} ([h(y^+, \hat{y}, n)], w)_\varepsilon = 0 \quad \forall (v, w) \in V \times W, \quad (29)$$

or

$$\sum_{\kappa \in \mathcal{T}} ((h(y^+, \hat{y}, n), v^+)_{\partial\kappa} - (\mathcal{F}(y), \nabla v)_\kappa) - \sum_{\varepsilon \in \mathcal{E}} ([h(y, \hat{y}, n)], w)_\varepsilon = 0 \quad \forall (v, w) \in V \times W. \quad (30)$$

The weak formulation (26) is recovered by the non-standard choice of numerical flux

$$h(y^+, \hat{y}, n) = n \cdot \mathcal{F}(y^+). \quad (31)$$

3.1.4 State equation and its linearization

We can represent (25) abstractly as a state equation $e(y) = 0$, by defining the state operator $e : Y \rightarrow (V \times W)^*$ for $y \in Y$ as

$$e(y) = (v, w) \mapsto \sum_{\kappa \in \mathcal{T}} (\nabla \cdot \mathcal{F}(y), v)_\kappa - \sum_{\varepsilon \in \mathcal{E}} ([n \cdot \mathcal{F}(y)], w)_\varepsilon, \quad (32)$$

The Fréchet derivative of the state operator (32) about a reference state y is then defined for a given perturbation $\delta y \in Y$ as

$$e'(y) \delta y = (v, w) \mapsto \sum_{\kappa \in \mathcal{T}} (\nabla \cdot \mathcal{F}'(y) \delta y, v)_\kappa - \sum_{\varepsilon \in \mathcal{E}} ([n \cdot \mathcal{F}'(y) \delta y], w)_\varepsilon. \quad (33)$$

3.2 Formulation in reference space with variable geometry

3.2.1 Mapping from reference space

In order to align discrete grid interfaces with flow interfaces, the grid must be treated as a variable. Therefore, we transform the strong formulation (9), (10) and weak formulation (25) of the flow equations in reference coordinates, in order to facilitate differentiation with respect to geometry. Therefore, we assume that there is a continuous, invertible mapping

$$u : \hat{\Omega} \rightarrow \Omega, \quad (34)$$

from a reference domain $\hat{\Omega} \subset \mathbb{R}^n$ to the physical domain $\Omega \subset \mathbb{R}^n$. We assume that $\hat{\Omega}$ is partitioned by $\hat{\mathcal{T}}$, so that $\bar{\hat{\Omega}} = \cup_{\hat{\kappa} \in \hat{\mathcal{T}}} \bar{\hat{\kappa}}$. Also, we consider the set of interfaces $\hat{\mathcal{E}}$ consisting of disjoint interfaces $\hat{\kappa}$, such that $\cup_{\hat{\kappa} \in \hat{\mathcal{E}}} \hat{\kappa} = \cup_{\hat{\kappa} \in \hat{\mathcal{T}}} \partial \hat{\kappa}$. The mapping u is further assumed to be (piecewise) differentiable with derivative or Jacobian matrix denoted

$$\nabla u|_{\hat{\kappa}} : \hat{\kappa} \rightarrow \mathcal{L}(\mathbb{R}^n, \mathbb{R}^n) \quad \forall \hat{\kappa} \in \hat{\mathcal{T}}. \quad (35)$$

The cofactor matrix $\text{cof}(\nabla u)|_{\hat{\kappa}} : \hat{\kappa} \rightarrow \mathcal{L}(\mathbb{R}^n, \mathbb{R}^n)$, is defined for $\hat{\kappa} \in \hat{\mathcal{T}}$,

$$\text{cof}(\nabla u(\hat{x})) = \det(\nabla u(\hat{x})) (\nabla u(\hat{x}))^{-\top} \quad \forall \hat{x} \in \hat{\kappa}. \quad (36)$$

Consider $\hat{\kappa} \in \hat{\mathcal{E}}$, for which we assume that there exists a parameterization $\theta_{\hat{\kappa}} : \hat{D} \rightarrow \hat{\kappa}$, mapping from a parameter space $\hat{D} \subset \mathbb{R}^{n-1}$. A parameterization of $\varepsilon = u(\hat{\kappa})$ is then given by the composition $\theta_{\varepsilon} = u \circ \theta_{\hat{\kappa}} : \hat{D} \rightarrow \varepsilon$. Given $\hat{\kappa} \in \hat{\mathcal{E}}$, the (non-unit) normal $s(\nabla u)|_{\hat{\kappa}} : \hat{\kappa} \rightarrow \mathbb{R}^n$ is defined for $\hat{x} \in \hat{\kappa}$ as the (non-unit) normal of the tangent plane of ε corresponding to the parameter $\theta_{\hat{\kappa}}^{-1}(\hat{x})$. For example, if $n = 3$, and ξ, η denotes the parametric coordinate directions, then for $\hat{x} \in \hat{\kappa}$

$$s(\nabla u)|_{\hat{\kappa}} = (\partial_{\xi} \theta_{\varepsilon} \wedge \partial_{\eta} \theta_{\varepsilon}) \circ \theta_{\hat{\kappa}}^{-1} \quad (37)$$

where $\partial_{\xi} \theta_{\varepsilon} \wedge \partial_{\eta} \theta_{\varepsilon}$ is the vector (or cross) product of the tangent plane basis vectors. A general formula for evaluating the vector product of tangent plane basis vectors is given by the following: let $(\mathbf{x}_1, \dots, \mathbf{x}_n)$ denote the coordinate directions in \mathbb{R}^n , and the parameterization be given in terms of components $\theta_{\varepsilon} = (\theta_{\varepsilon}^1, \dots, \theta_{\varepsilon}^n)$, then

$$\partial_{\xi_1} \theta_{\varepsilon} \wedge \dots \wedge \partial_{\xi_{n-1}} \theta_{\varepsilon} = \det \begin{pmatrix} \mathbf{x}_1 & \dots & \mathbf{x}_n \\ \partial_{\xi_1} \theta_{\varepsilon}^1 & \dots & \partial_{\xi_1} \theta_{\varepsilon}^n \\ \vdots & \ddots & \vdots \\ \partial_{\xi_{n-1}} \theta_{\varepsilon}^1 & \dots & \partial_{\xi_{n-1}} \theta_{\varepsilon}^n \end{pmatrix}, \quad (38)$$

so that in general

$$s(\nabla u)|_{\hat{\kappa}} = (\partial_{\xi_1} \theta_{\varepsilon} \wedge \dots \wedge \partial_{\xi_{n-1}} \theta_{\varepsilon}) \circ \theta_{\hat{\kappa}}^{-1}, \quad (39)$$

where we assume also that $\|\partial_{\xi_1} \theta_{\varepsilon} \wedge \dots \wedge \partial_{\xi_{n-1}} \theta_{\varepsilon}\| = 1$.

3.2.2 Geometric boundary conditions

In order to implement geometric boundary conditions, which constrain points to the boundary, we introduce a nonlinear projection operator, denoted $b : U \rightarrow U$. A common and simple example of such a mapping would be for the case of constraining points to a boundary surface $\varepsilon \in \mathcal{E}_{\partial}$, which lies on a hyperplane

$$P = \{x \in \mathbb{R}^n \mid x \cdot n_P = x_P \cdot n_P\}, \quad (40)$$

with a given unit normal n_P and reference point x_P . In this case, we define the projection to map the point to the closest point on the hyperplane

$$b(u)|_{\hat{\kappa}} = u - (n_P \cdot (u - x_P)) n_P. \quad (41)$$

Another example is the projection of a point to a fixed point x_0 , for example, the point lying at the intersection of n hyperplanes in \mathbb{R}^n ,

$$b(u)|_{\hat{\epsilon}} = x_0, \quad (42)$$

so that in particular the derivative of the projection vanishes at such points.

3.2.3 Strong and weak formulation in reference space

The strong formulation in reference coordinates is

$$(\text{cof}(\nabla u) \nabla) \cdot \mathcal{F}(y) = 0 \text{ in } \hat{\kappa} \quad \forall \hat{\kappa} \in \hat{\mathcal{T}}, \quad (43)$$

$$[\![s(\nabla u) \cdot \mathcal{F}(y)]\!] = 0 \text{ on } \hat{\epsilon} \quad \forall \hat{\epsilon} \in \hat{\mathcal{E}}, \quad (44)$$

$$b(u) - u = 0 \text{ on } \hat{\epsilon} \quad \forall \hat{\epsilon} \in \hat{\mathcal{E}}_\partial. \quad (45)$$

We assume that $Y = [H^k(\hat{\mathcal{T}})]^m$ now consists of functions defined in an \mathbb{R}^m -valued broken Sobolev space over $\hat{\mathcal{T}}$, where k is a positive integer. We also assume that $U = [H^\ell(\hat{\Omega})]^n$, the \mathbb{R}^n -valued Sobolev space over $\hat{\Omega}$. We further assume that the test spaces $V = Y$ and W , the corresponding single-valued trace space, now consist of functions defined over reference space. We then define a provisional state operator $\tilde{e} : Y \times U \rightarrow (V \times W)^*$ for $(y, u) \in Y \times U$,

$$\tilde{e}(y, u) = (v, w) \mapsto \sum_{\hat{\kappa} \in \hat{\mathcal{T}}} ((\text{cof}(\nabla u) \nabla) \cdot \mathcal{F}(y), v)_{\hat{\kappa}} - \sum_{\hat{\epsilon} \in \hat{\mathcal{E}}} ([\![s(\nabla u) \cdot \mathcal{F}(y)]\!], w)_{\hat{\epsilon}}, \quad (46)$$

which has a Fréchet derivative defined for perturbation $(\delta y, \delta u) \in Y \times U$, and test functions $(v, w) \in V \times W$, by

$$\begin{aligned} \langle \tilde{e}'(y, u)(\delta y, \delta u), (v, w) \rangle &= + \sum_{\hat{\kappa} \in \hat{\mathcal{T}}} (((\text{cof}'(\nabla u) \nabla \delta u) \nabla) \cdot \mathcal{F}(y), v)_{\hat{\kappa}} \\ &\quad + \sum_{\hat{\kappa} \in \hat{\mathcal{T}}} ((\text{cof}(\nabla u) \nabla) \cdot (\mathcal{F}'(y) \delta y), v)_{\hat{\kappa}} \\ &\quad - \sum_{\hat{\epsilon} \in \hat{\mathcal{E}}} ([\![s'(\nabla u) \nabla \delta u] \cdot \mathcal{F}(y)]\!, w)_{\hat{\epsilon}} \\ &\quad - \sum_{\hat{\epsilon} \in \hat{\mathcal{E}}} ([\![s(\nabla u) \cdot (\mathcal{F}'(y) \delta y)]\!], w)_{\hat{\epsilon}}. \end{aligned} \quad (47)$$

The state operator $e : Y \times U \rightarrow (V \times W)^*$, which imposes geometric boundary conditions (45) by composing the provisional state operator (46) with the projection $b(u)$, is then defined

$$e(y, u) = \tilde{e}(y, b(u)), \quad (48)$$

with Fréchet derivative defined, for state $(y, u) \in Y \times U$ and perturbation $(\delta y, \delta u) \in Y \times U$, by

$$\langle e'(y, u)(\delta y, \delta u) \rangle = \tilde{e}_y(y, b(u)) \delta y + \tilde{e}_u(y, b(u)) b'(u) \delta u. \quad (49)$$

The state equation in reference coordinates is $e(y, u) = 0$. The corresponding weak formulation in reference coordinates is: find $(y, u) \in Y \times U$ such that

$$\langle e(y, u), (v, w) \rangle = 0 \quad \forall (v, w) \in V \times W. \quad (50)$$

4. SOLVER

In this section, we present an approach for solving the MDG-ICE weak formulation in reference coordinates (50), including discretization using standard polynomial finite elements, the nonlinear solver strategy, and the grid operations necessary for modifying the grid topology as the solver degenerates elements.

4.1 Discretization

Discretization of the weak formulation (50) implies choosing finite-dimensional subspaces $Y_h \subset Y$, $U_h \subset U$, $V_h \subset V$ and $W_h \subset W$. We use standard piecewise polynomials, cf. [9], defined over reference elements. Let \mathcal{P}_p denote the space of polynomials spanned by the monomials \mathbf{x}^α with multi-index $\alpha \in \mathbb{N}_0^n$, satisfying $\sum_{i=1}^n \alpha_i \leq p$. In the case of a simplicial grid,

$$Y_h = \left\{ y \in Y \mid \forall \hat{\kappa} \in \hat{\mathcal{T}}, y|_{\hat{\kappa}} \in [\mathcal{P}_p]^m \right\}. \quad (51)$$

Let \mathcal{Q}_p denote the space of polynomials spanned by the monomials \mathbf{x}^α with multi-index $\alpha \in \mathbb{N}_0^n$, satisfying $\alpha_i \leq p$ for $i = 1 \dots n$. In the case of a cuboid grid,

$$Y_h = \left\{ y \in Y \mid \forall \hat{\kappa} \in \hat{\mathcal{T}}, y|_{\hat{\kappa}} \in [\mathcal{Q}_p]^m \right\}. \quad (52)$$

The discrete subspace U_h of mappings from reference space to physical space are also discretized into \mathbb{R}^n -valued piecewise polynomials, in the case of a simplicial grid

$$U_h = \left\{ u \in U \mid \forall \hat{\kappa} \in \hat{\mathcal{T}}, u|_{\hat{\kappa}} \in [\mathcal{P}_p]^n \right\}, \quad (53)$$

or in the case of a cuboid grid

$$U_h = \left\{ u \in U \mid \forall \hat{\kappa} \in \hat{\mathcal{T}}, u|_{\hat{\kappa}} \in [\mathcal{Q}_p]^n \right\}. \quad (54)$$

The case that the chosen polynomial degree of U_h is equal to that of Y_h is referred to as isoparametric. It is also possible to choose the polynomial degree of U_h to be less (sub-parametric) or greater (super-parametric) than that of Y_h .

The test space V_h is chosen to be U_h for a Galerkin-type method. W_h is the subspace of single-valued polynomials defined over the interface. In the case of a simplicial grid

$$W_h = \left\{ w \in W \mid \forall \hat{\epsilon} \in \hat{\mathcal{E}}, w|_{\hat{\epsilon}} \in [\mathcal{P}_p]^m \right\}, \quad (55)$$

while in the case of a cuboid grid

$$W_h = \left\{ w \in W \mid \forall \hat{\mathcal{E}} \in \hat{\mathcal{E}}, w|_{\hat{\mathcal{E}}} \in [Q_p]^m \right\}. \quad (56)$$

The discrete weak formulation is therefore just (50), only restricted to the discrete subspaces,

$$e_h : Y_h \times U_h \rightarrow V_h \times W_h. \quad (57)$$

4.2 Nonlinear solver

Since in general $\dim(Y_h) + \dim(U_h) \neq \dim(V_h) + \dim(W_h)$, instead of solving

$$e_h(y, u) = 0 \quad (58)$$

directly, we instead seek a stationary point

$$e'_h(y, u) = 0. \quad (59)$$

The unconstrained optimization problem (59) can be solved iteratively. Given an initialization $(y, u)_0$ the solution is repeatedly updated

$$(y, u)_{i+1} = (y, u)_i + \Delta(y, u)_i \quad i = 0, 1, 2, \dots \quad (60)$$

until (59) is satisfied to a given tolerance.

Available iterative methods for solving (59) include the first-order gradient descent method with increment given by

$$\Delta(y, u) = -\alpha e'_h(y, u)^* e_h(y, u), \quad (61)$$

the second-order Newton method with increment given by

$$\Delta(y, u) = -e''_h(y, u)^{-1} e'_h(y, u), \quad (62)$$

or the Gauss-Newton method with increment given by

$$\Delta(y, u) = -\left(e'_h(y, u)^* e'_h(y, u)\right)^{-1} \left(e'_h(y, u)^* e_h(y, u)\right). \quad (63)$$

As the residual approaches zero, the Gauss-Newton method recovers the second order convergence rate of the Newton method (62), while avoiding difficulties such as a loss of positive definiteness. Other choices are possible, such as the Levenberg-Marquardt method,

$$\Delta(y, u) = -\left(e'_h(y, u)^* e'_h(y, u) + \lambda I(y, u)\right)^{-1} \left(e'_h(y, u)^* e_h(y, u)\right). \quad (64)$$

where $I(y, u)$ denotes the identity mapping and $\lambda > 0$ denotes a regularization factor, that ensures a full rank linear system of equations. This method, can be viewed as an adaptive combination of the gradient descent (61) and Gauss-Newton (63) methods, where the former is recovered by taking λ large, while the latter is recovered as $\lambda \rightarrow 0$. Another possible choice would be the Lagrange-Newton method for constrained optimization [57].

In the present work, we use the Levenberg-Marquardt method (64), which requires solving a symmetric, positive definite system of linear system of equations. The regularization $\lambda I(y, u)$ in (64) can be generalized to provide finer-grained control of the regularization with respect to different solution variables. Since the system (59) is full rank with respect to y , a small or zero regularization can be applied with respect to y , partially recovering the Gauss-Newton method (63). On the other hand, the system (59) is often of insufficient rank with respect to geometry, so that a greater regularization can be applied with respect to u , especially for the initial nonlinear solver iterations. This regularization is also useful for inhibiting the solver from modifying the discrete geometry u too aggressively. Therefore, the regularization we use is in the present work is of the form $(y, u) \mapsto (\lambda_y y, \lambda_u u)$, with $\lambda_y, \lambda_u > 0$.

Such a positive definite, symmetric system of equations can be solved for using either CG (conjugate gradients) or MINRES [58]. Other available least squares solvers include LSQR [59] and LSMR [60]. The linear systems of equations are in general poorly conditioned, especially when the regularization parameter is low, so that preconditioning is essential. In the present work, we employ MINRES using right preconditioning with a diagonal preconditioner [61].

4.3 Grid Operations

While in certain situations (cf. Section 5.1) the MDG-ICE solver is able to solve directly for the grid, without modifying the grid topology, in general an arbitrary initial grid topology will not be compatible with the interface topology present in the exact solution (cf. Section 5.4). In such situations MDG-ICE drives cells in the discrete geometry u to degenerate, which can be identified by their generating a negative Jacobian, $\det \nabla u < 0$. In such situations, degenerate simplicial cells can be removed using standard local grid operations including edge collapse, cf. [62], while in order to maintain a sufficient level of refinement, remaining cells can be refined to maintain a target number of cells, or to ensure that a maximum length scale is not exceeded. In the case of cuboid cells, when a cell degenerates, we implement partial collapses by aliasing degrees of freedom on opposite faces of the cell, an example of the application of this type of degeneration is given in Section 5.2.

Upon convergence of the MDG-ICE solver to a stationary point (59), if the residual (58) is not yet converged within a given tolerance, then the grid can be refined and a new stationary point sought on a finer space (59).

4.4 Entropy Condition Enforcement

For nonlinear convection problems, in order to ensure uniqueness, an entropy inequality must be enforced, cf. [63]. In the context of piecewise smooth flow, this requires that

$$[n \cdot \mathcal{F}^s(y)] \geq 0 \text{ on } \varepsilon \quad \forall \varepsilon \in \mathcal{E}, \quad (65)$$

where her $\mathcal{F}^s(y)$ denotes an entropy flux. In the case of inviscid compressible flow, governed by the Euler equations, the spatial entropy flux can be defined as

$$\mathcal{F}^s(y) = (\rho s v_1, \dots, \rho s v_n) \in \mathbb{R}^n, \quad (66)$$

while the spacetime entropy flux can be defined as

$$\mathcal{F}^s(y) = (\rho s v_1, \dots, \rho s v_n, \rho s) \in \mathbb{R}^{n+1}, \quad (67)$$

where the entropy variable, which is negated in relation to physical entropy, can be defined as

$$\rho s = -\frac{\rho}{\gamma-1} \ln(p/\rho^\gamma). \quad (68)$$

In the present work, a very simple restart strategy is employed for the preliminary test cases considered in Section 4.4. Between each nonlinear solver iteration, along any interface for which the entropy condition is violated, the states are averaged. Since entropy condition violating solutions are dynamically unstable, this approach appears to be sufficient for the preliminary test cases considered in the present work, as once an entropy-violating discontinuity is broken, the solver is not strongly driven to recreate it. For more complex flows, a more elaborate entropy condition enforcement may be required, which will be investigated in future work.

5. EXAMPLES

We now use MDG-ICE to compute a variety of benchmark flow configurations, in order to provide an initial assessment of its ability to stably and accurately compute piecewise smooth flows with discontinuous interfaces. The first problem, presented in Section 5.1, a one-dimensional, unsteady, moving interface, is solved for the case of both a small and large motion. This case illustrates the ability of MDG-ICE to solve for flows with interfaces with a minimal reliance on grid topology modification. While in the first problem, the interface is imposed as a temporal inflow condition, the second problem, presented in Section 5.2, is that of shock formation. This problem assesses the ability of MDG-ICE to solve for flows with grid topology that changes in time. The third problem, presented in Section 5.3, the Sod shock tube problem, assesses the ability of MDG-ICE to handle a number of different types of waves within a unified formulation. These include a shock and contact discontinuity, as well as a rarefaction, that introduces both a singularity and derivative discontinuities. The fourth problem, presented in Section 5.4, is that of intersecting oblique shocks, which assesses the ability of MDG-ICE, in conjunction with local grid operations, to compute a priori unknown interfaces in both two-dimensional and three-dimensional space. The fifth problem, presented in Section 5.5, of supersonic flow over a cylinder, assesses the ability of MDG-ICE to compute a priori unknown curved interface geometry, while retaining a high-order representation up to the interface.

5.1 Moving Interface

First, we revisit the case involving a moving interface governed by the spacetime linear advection equation with spatial velocity $v_x = \frac{1}{10}$ considered in Section 1.1. Figure 2 depicts the solution computed using

MDG-ICE. The solver was initialized with a uniform grid, with the temporal initial condition (3) at $t = 0$ extruded in time. MDG-ICE computes the exact solution, and in particular, fits the interface present in the exact solution, which is illustrated with a dashed line. This piecewise constant flow was computed for the cases of $p = 0, 1, 2, 3$, in order to demonstrate that MDG-ICE avoids the artificial dissipation introduced by low-order methods with unfitted interfaces, while also avoiding the oscillations produced by higher-order methods in the presence of unfitted interfaces, as shown in Figure 1 for the case of the standard DG method. An analogous flow was computed using a larger spatial velocity $v_x = \frac{3}{2}$ using MDG-ICE($p = 0$), as shown in Figure 3. In this case, the upstream state transitioned across many cells, including around one of the corners of the domain to reproduce the exact solution, without requiring any grid topology changes. This case demonstrates that MDG-ICE is also able to fit interfaces with large motion by simultaneously updating the flow field and the grid, with minimal reliance on grid topology modification.

5.2 Spacetime Burgers flow

In the previous example, an interface was specified as an initial condition. An interesting question, is whether or not a method for treating interfaces can handle changes in topology in time, which happens, for example, in the case of shock formation. An additional related phenomenon that occurs in inviscid flows, is that of singularities, which can occur just prior to shock formation. To assess the ability of MDG-ICE to compute such flows, we now consider a shock formation problem governed by the spacetime Burgers equation with flux

$$\mathcal{F}(y) = \left(\frac{1}{2}y^2, y \right). \quad (69)$$

The smooth initial condition at $t = 0$ is specified as

$$y(x, t = 0) = \begin{cases} 2 & x < \frac{1}{4} \\ 4 - 8x & \frac{1}{4} \leq x \leq \frac{3}{4} \\ -2 & x > \frac{3}{4} \end{cases}, \quad (70)$$

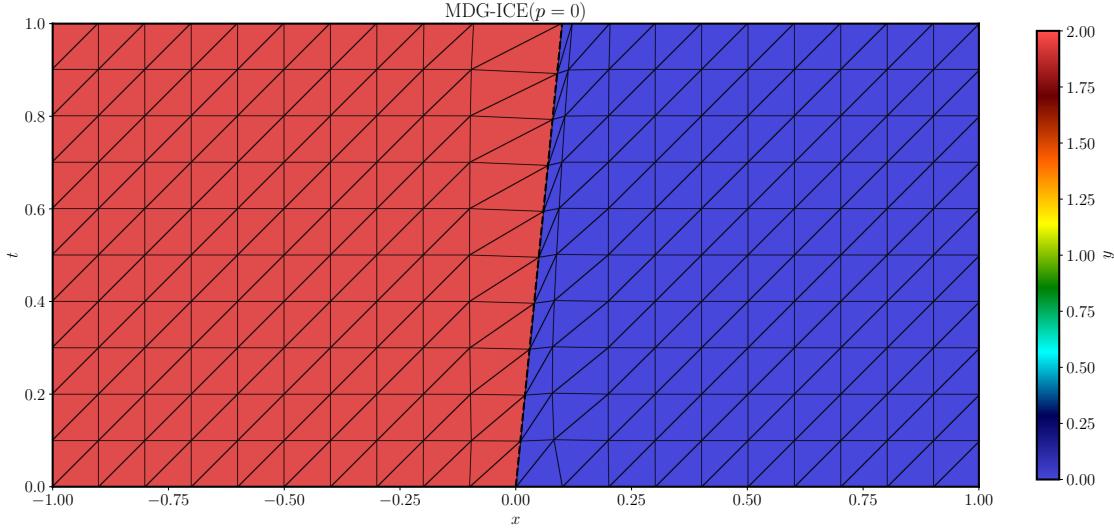
which leads to a shock at $t = \frac{1}{8}$. The exact spacetime solution is

$$y(x, t) = \begin{cases} 2 & x < x^+(t) \\ \frac{8x-4}{8t-1} & x^+(t) \leq x \leq x^-(t) \\ -2 & x > x^-(t) \end{cases}, \quad (71)$$

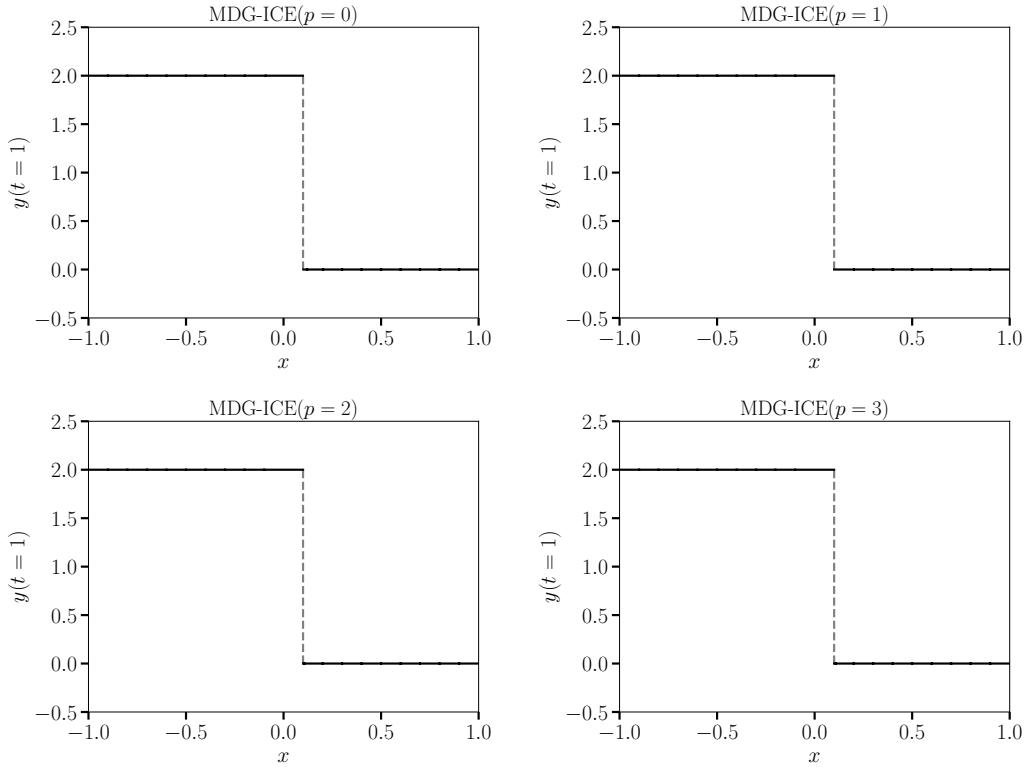
where the shock positions are denoted

$$x^+(t) = \min \left(\frac{1}{4} + 2t, \frac{1}{2} \right), \quad (72)$$

$$x^-(t) = \max \left(\frac{3}{4} - 2t, \frac{1}{2} \right). \quad (73)$$



(a) The converged spacetime solution obtained for $v_x = \frac{1}{10}$. The solver was initialized with a uniform grid, with the initial condition at $t = 0$ extruded in time. The solver was initialized with a uniform grid, with the initial condition at $t = 0$ extruded in time. Upon convergence, the interface present in the exact solution, illustrated with a dashed line, has been fit.



(b) The converged piecewise polynomial solutions obtained for MDG-ICE($p = 0, 1, 2, 3$), sampled along the line $t = 1$. The dashed line located at $x = \frac{1}{10}$ spans the interface present in the exact solution.

Fig. 2: A one-dimensional moving interface, computed using MDG-ICE.

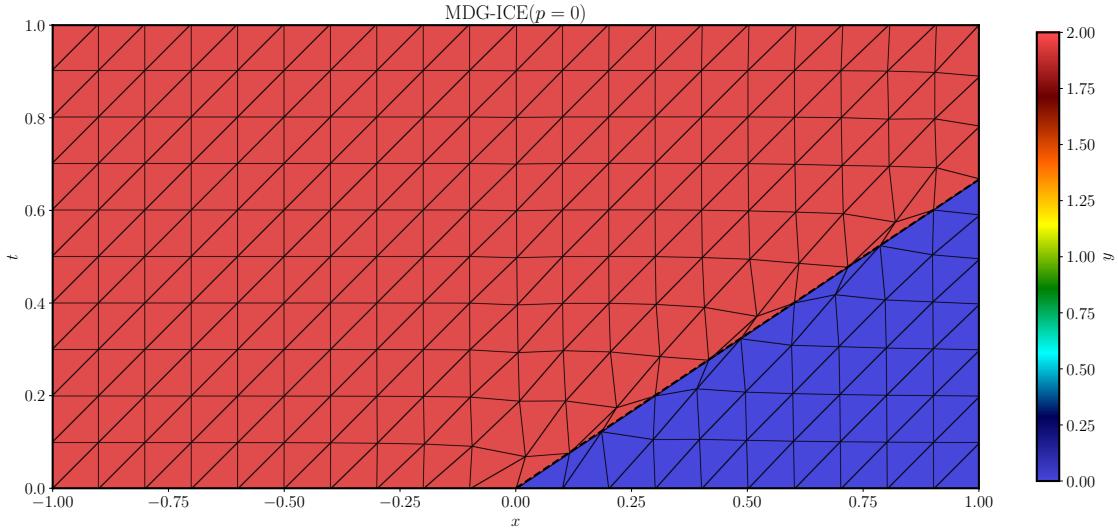


Fig. 3: A one-dimensional moving interface, with a velocity of $v_x = \frac{3}{2}$, shown in a two-dimensional spacetime domain, computed using MDG-ICE. The solver was initialized with a uniform grid, with the initial condition at $t = 0$ extruded in time. Upon convergence, the interface present in the exact solution, illustrated with a dashed line, has been fit. The interface exits the domain at $t = \frac{2}{3}$.

This is a variant of the problem studied by Giles and Ulbrich [52–54] when considering adjoint solutions for shocked flows.

The solution computed using MDG-ICE($p = 2$) on a grid discretizing the spacetime domain $\Omega = (0, 1) \times (0, \frac{1}{2})$ with six quadrilateral cells is shown in Figure 4. The flow state variable y was discretized using \mathcal{Q}_2 elements, while the shape variable u was discretized using \mathcal{Q}_1 elements. Figure 4a shows the solution initialization, which was the extrusion of the initial condition (70) in time. With only six cells, MDG-ICE is able to reproduce the exact solution by partially degenerating one cell, while fully degenerating another cell. The degeneration process involves imposing an equality constraint on initially distinct degrees of freedom, so that upon subsequent solver iterations they move in a synchronized fashion. The partially degenerated cell has its top two points collapsed, enabling it to exactly reproduce the singularity leading into the point of shock formation. As shown in Figure 4c, the flow is piecewise linear in space, with an increasing slope that transitions to an instantaneous jump at $(x, t) = (\frac{1}{2}, \frac{1}{8})$. Emanating from this point is the fully degenerated cell, which has its top two and bottom two points collapsed, coinciding with the shock interface. This case indicates that MDG-ICE is capable of handling interface topology that changes in time as well as flows with singularities.

5.3 Shock tube

In this example, we consider a shock tube problem, governed by the spacetime Euler equations (23), where multiple interfaces emanate from a discontinuity at $t = 0$. The initial conditions are those of the

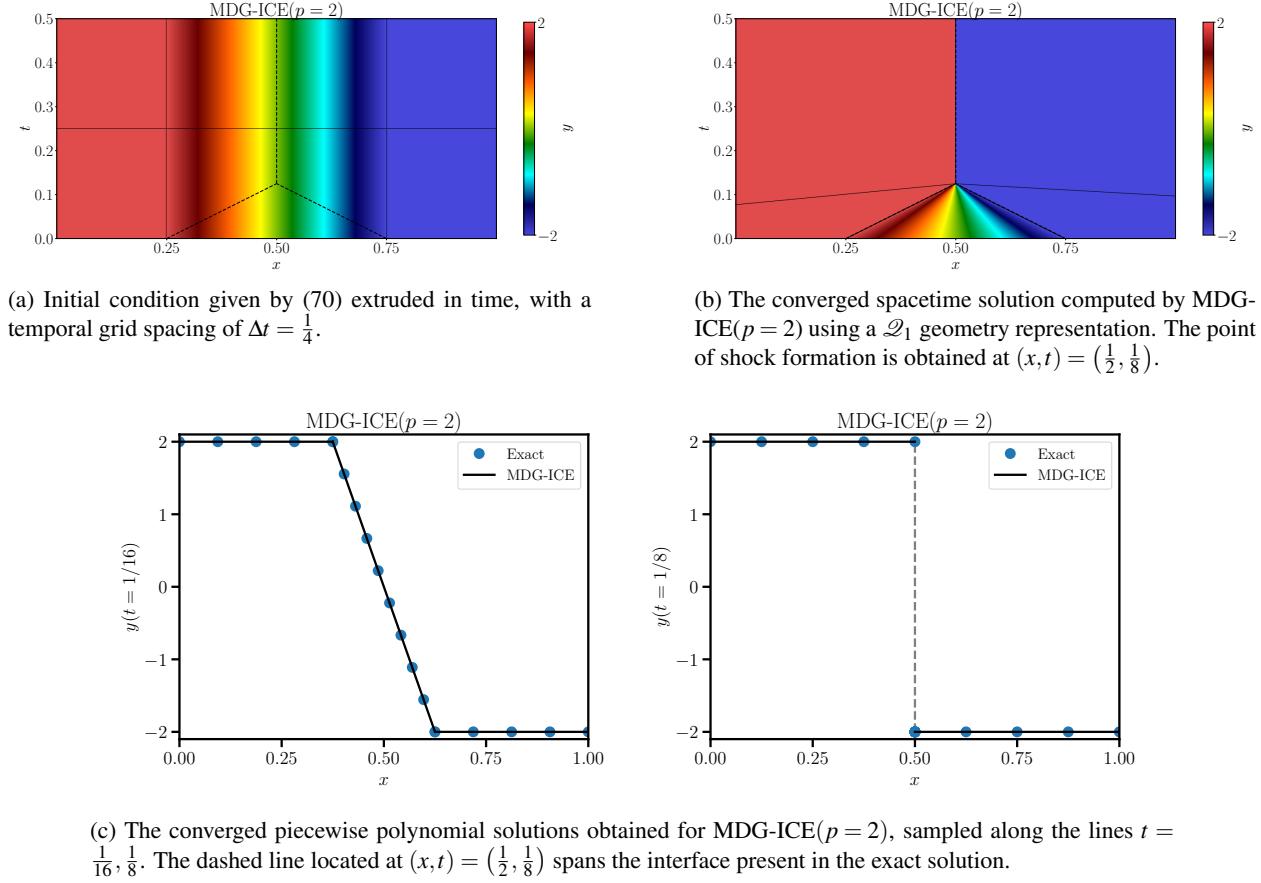


Fig. 4: Burgers spacetime shock formation

classic Sod shock tube, with a piecewise constant state, with

$$(\rho, v, p)(x, t = 0) = \begin{cases} (1, 0, 1) & x < 0, \\ (\frac{1}{8}, 0, \frac{1}{10}) & x > 0, \end{cases} \quad (74)$$

The solution computed using MDG-ICE($p = 3$) on a grid discretizing the spacetime domain $\Omega = (-\frac{1}{2}, \frac{1}{2}) \times (0, \frac{2}{10})$ with ten quadrilateral cells is shown in Figure 5 within a single spacetime slab. The flow state variable y was discretized using \mathcal{Q}_3 elements, while the shape variable u was discretized using \mathcal{Q}_1 elements. Figure 5a shows the spacetime initialization, which was the extrusion of the initial condition (74) in time. Also, the middle eight cells were degenerated to the point $(x, t) = 0$ at the temporal initialization. Figures 5c-5f show a comparison of density, pressure, velocity, and entropy between the MDG-ICE solution and the exact solution, cf. [64]. We observe in Figure 5c that both the shock and contact are fitted as true discontinuities, at the correct location, indicating that the speed of each is computed correctly. Moreover at the head and tail of the rarefaction, a derivative discontinuity exists, which is also fit. Each of these types of discontinuities were computed without resorting to specialized logic specific to each type. This highlights one of the key features of MDG-ICE, that multiple types of discontinuities can be detected and fitted simply by solving

the weak formulation (50), which enforces both the conservation law (9) and its interface condition (10) separately. Another key observation is that even though MDG-ICE does not rely on artificial dissipation for stability, the computed solutions are oscillation-free while in Figure 5f we observe the absence of spurious entropy generation at the shock.

5.4 Intersecting oblique shocks

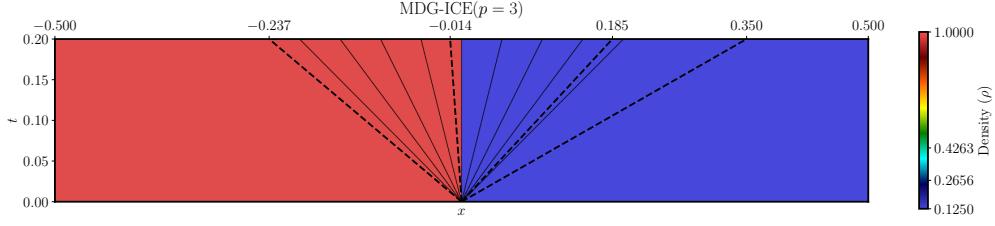
State	Pressure (p)	Mach (M)	Density (ρ)	Entropy (s)
1	1	2	1	1.177652828
2	1.31540694	1.821253900	1.21557716	1.179751529
3	1.70550887	1.648699723	1.46260855	1.181536318
4	2.187780929	1.478144666	1.74655696	1.183109810

Table 1: Oblique shocks: Pressure, Mach, Density and Entropy values computed at each of the distinct states in the intersecting oblique shock flow, shown in Figures 6, 7, and 8 in 2D, and Figure 9 in 3D.

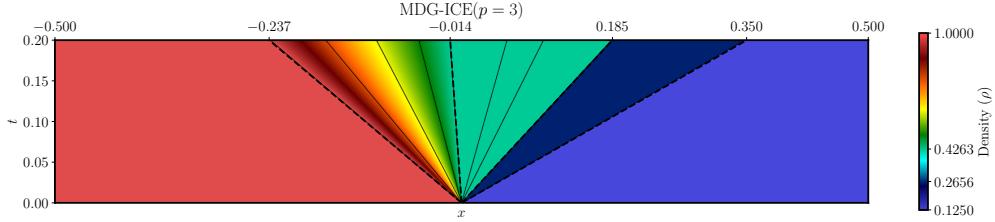
In this example, we consider initially unfitted intersecting oblique shocks, governed by the steady Euler equations (22), in order to assess the ability of MDG-ICE to compute flows with non-trivial and a-priori unknown interface topology. The inflow condition are specified with a Mach number $M_1 = 2$, density $\rho_1 = 1$, and pressure $p_1 = 1$. The domain and grids used for simulating this flow are shown in Figure 6.

First we computed this flow using the standard $\text{DG}(p = 1)$ method with the HLLC flux [64, 65], augmented with a shock capturing strategy based on [16]. The coarse grid with 641 linear triangular cells shown in Figure 6a was used to compute the density field shown in Figure 7a. In order to reduce the artificial dissipation present in the solution, we employed a residual-based adaptation strategy to refine the grid locally, shown in Figure 6b, resulting in reduced artificial dissipation around the shocks as shown in Figure 7b. We also see that the considerable amount of spurious entropy generation produced by shock capturing on the coarse grid, Figure 8a, was reduced by adaptive grid refinement as shown in Figure 8b, but that it is still significant.

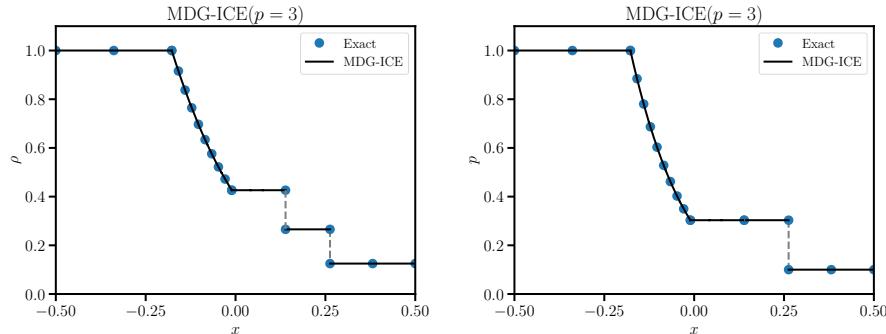
Next we computed this flow using $\text{MDG-ICE}(p = 1)$, which does not rely on any form of shock capturing or artificial dissipation. We initialized the solution using the coarse grid $\text{DG}(p = 1)$ shock-captured solution on 641 linear triangle cells (Figure 7a). As the MDG-ICE solver progressed, any cells that degenerated were removed from the grid via local collapse operations, as described in Section 4.3. The resulting grid, shown in Figure 6c, had 461 linear triangle cells remaining, with the intersecting shock interfaces clearly visible. The corresponding density field, shown in Figure 7c, shows that the solver has fit the shocks as true discontinuities with an instantaneous jump across each shock interface. For this piecewise constant flow, the MDG-ICE residual has vanished, indicating that the computed solution is exact. The computed flow values, for each flow field state are listed in Table 1. As in the previous example, the entropy field, shown in Figure 8c, indicates that no spurious entropy is generated.



(a) The initial spacetime grid and density field over $\Omega = (-0.5, 0.5) \times (0, 0.2)$. The initial solution consists of 10 \mathcal{Q}_3 quadrilateral cells using a \mathcal{Q}_1 shape representation, where the middle eight cells are degenerated at the point $(x, t) = 0$, as part of the temporal initialization. The dashed lines indicate the location of the shock, contact, and rarefaction head and tail that are present in the exact solution.

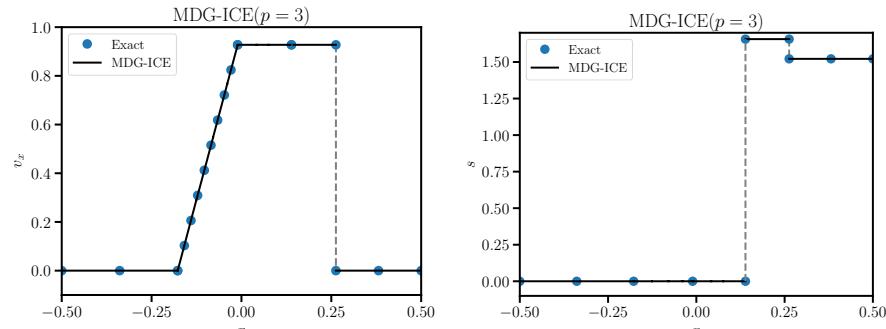


(b) The converged spacetime grid and density field over $\Omega = (-0.5, 0.5) \times (0, 0.2)$. The final solution consists of 10 \mathcal{Q}_3 quadrilateral cells using a \mathcal{Q}_1 shape representation. The dashed lines indicate the location of the shock, contact, and rarefaction head and tail that are present in the exact solution.



(c) Density at $t = 0.15$

(d) Pressure at $t = 0.15$



(e) Velocity at $t = 0.15$

(f) Entropy $s = (\gamma - 1)^{-1} \ln(p/\rho^\gamma)$ at $t = 0.15$

Fig. 5: Sod shock tube

Finally we computed three-dimensional reflecting planar shocks using MDG-ICE($p = 1$), with flow conditions that are analogous to those of the two-dimensional case. While, physically, the three-dimensional flow is equivalent to the two-dimensional flow, the grid is fully unstructured and moves in all dimensions. The initial grid consisted of 9,724 linear tetrahedral cells. A challenge that arises when computing three-dimensional fitted interfaces is that the grid topology has increased complexity, which can manifest in a greater tendency for cells to degenerate and collapse. The final grid, which consisted of 10,750 linear triangle cells, was obtained by MDG-ICE in conjunction with local collapse operations, to remove degenerate cells, along with local refinement, to refine cells that exceeded a maximum length scale of 0.1 in order to ensure that sufficient resolution was maintained. The computed flow field is shown in Figure 9. As in the two-dimensional case, the MDG-ICE residual has vanished, indicating that the computed solution is exact.

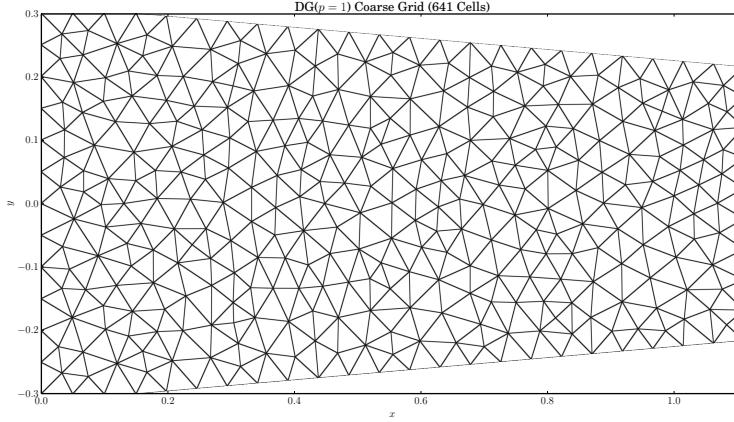
This case shows that MDG-ICE does not require a priori knowledge of the shock location and topology, or impose restrictions on the topology. As can be seen, the method handles shock-shock as well as shock-wall interactions. This demonstrates the ability of MDG-ICE to fit initially unfitted shocks, even with non-trivial topology. The three-dimensional intersecting planar oblique shocks, indicates that the ability of MDG-ICE to compute flows with non-trivial interface topology generalizes to higher dimensions.

5.5 Bow Shock

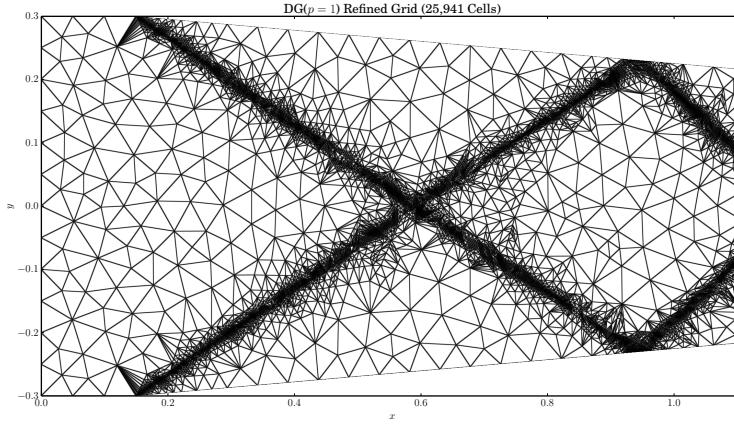
So far the problems considered have involved linear interface geometry. Since the MDG-ICE formulation makes no assumption about the shape representation, and accommodates standard finite element shape representations, we consider a problem involving curved boundaries and interface geometry. The problem is that of supersonic flow over a cylinder, resulting in a curved bow shock as described by Shu [66]. Here we also use an initial grid consisting of 15×20 points, or 14×19 cells, which we then decompose into 532 unstructured triangular cells. The inflow condition is Mach number $M_\infty = 3$, pressure $p_\infty = 1$, and density $\rho_\infty = 1.4$.

First we computed the solution using the standard DG($p = 1$) method with the HLLC flux [64, 65], augmented with a shock capturing strategy based on [16]. This shock captured solution was then used as an initialization to compute the flow field using MDG-ICE($p = 3$) with \mathcal{P}_2 -shape triangle elements that were constrained to the outer ellipse and inner circle via geometric boundary conditions. The grids are shown in Figure 10, along with the computed pressure fields in Figure 11. Since the exact solution has a constant stagnation enthalpy (20) throughout the domain, $H_\infty = 7$, this quantity is shown as well in Figure 12.

We observe that the curved bow shock is fit by the MDG-ICE solver in conjunction with local grid operations to collapse degenerate cells while performing local refinement to maintain a target number of 532 cells. The location of the shock along the line $y = 0$ was computed as $x = -1.698$ for a stand-off distance of 0.698. As a form of shock fitting that does not rely on any form of shock capturing or artificial dissipation, MDG-ICE appears to be immune to carbuncle instabilities [67, 68]. The computed pressure field also appears to be free of the detrimental effects of artificial dissipation that are apparent in the shock captured solution. Furthermore, MDG-ICE more closely preserves a constant value of stagnation enthalpy throughout the domain, reducing the stagnation enthalpy error $\|H - H_\infty\|_{L_2(\Omega)}$ nearly two orders of magnitude compared to the DG($p = 1$) solution, from 1.2633×10^{-1} to 1.9704×10^{-3} . The pressure and Mach number distribution, sampled along the line $y = 0$, along with the exact pressure at the stagnation point, $p \approx 12.06096$, are shown in Figure 13. The computed pressure and Mach number at the stagnation point on this coarse grid are 12.0559064 and $2.26753700 \times 10^{-2}$ respectively.



(a) An initial coarse grid of 641 cells.



(b) A finer grid obtained using isotropic, residual-based adaptation, resulting in 25,941 cells.

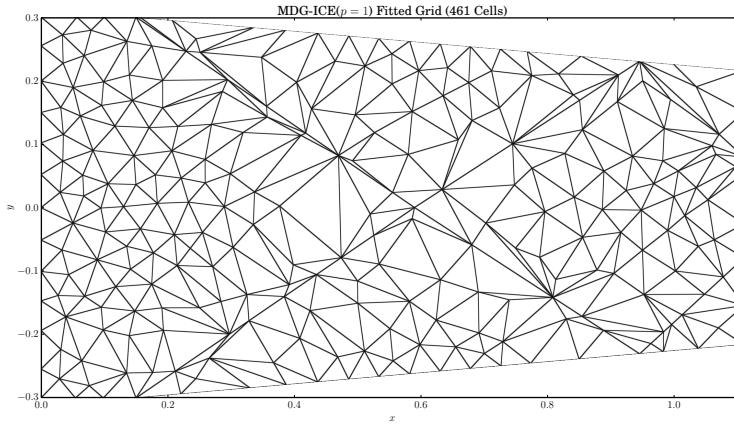
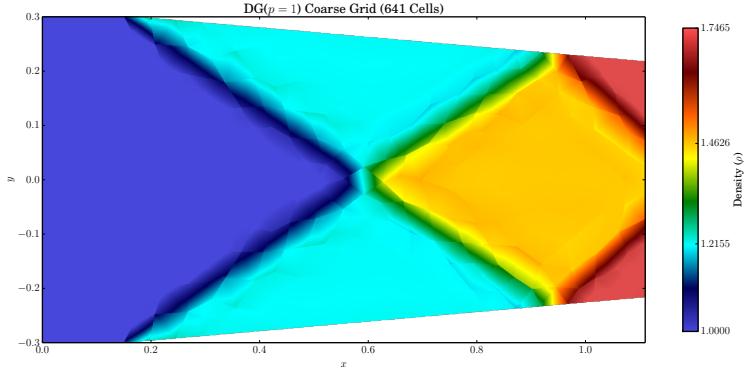
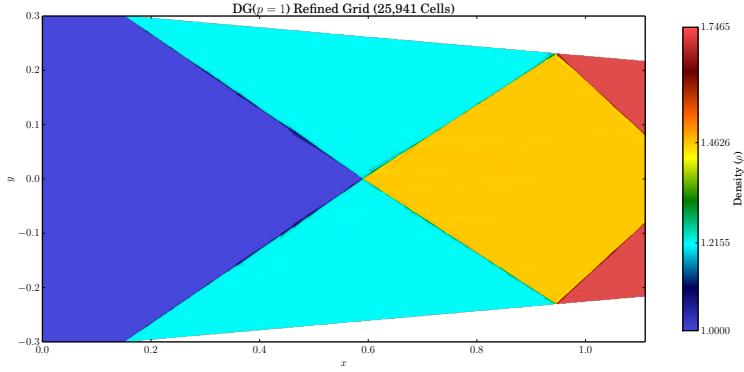
(c) A fitted grid computed by MDG-ICE($p = 1$) without shock capturing, with degenerate cell collapse, using the coarse grid shown in (6a) as the initialization, resulting in 461 cells. The shock interfaces present in the exact solution are reproduced by the grid.

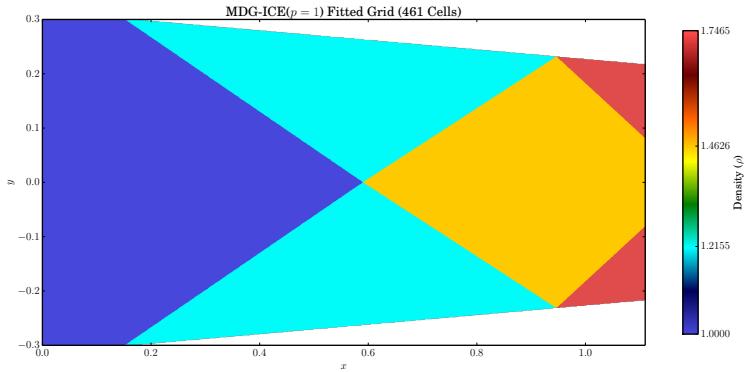
Fig. 6: Intersecting oblique shocks: Grid



(a) The density field computed using $\text{DG}(p = 1)$ with shock capturing on a coarse grid with 641 cells. The effect of artificial dissipation introduced by the shock capturing is noticeable.

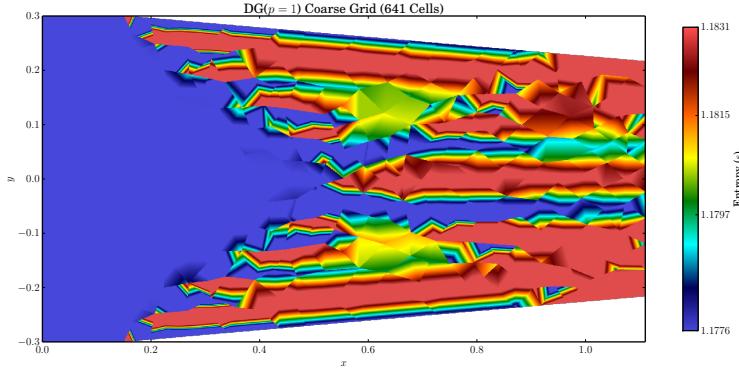


(b) The density field computed using $\text{DG}(p = 1)$ with shock capturing on the finer grid with 25,941 cells. The effect of artificial dissipation introduced by the shock capturing is reduced from its effect on the coarser grid, but is still visible.

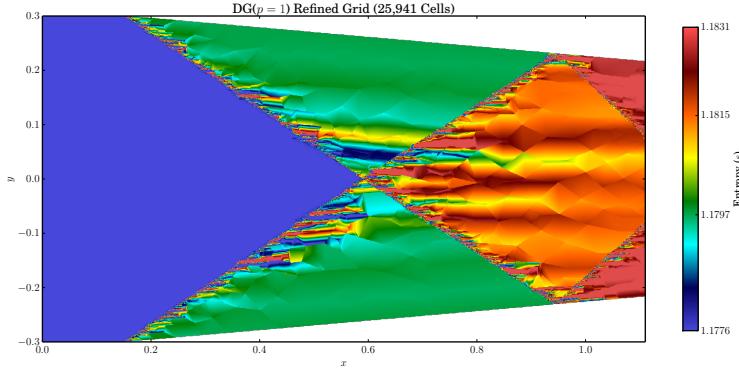


(c) The density field computed using $\text{MDG-ICE}(p = 1)$ without shock capturing.

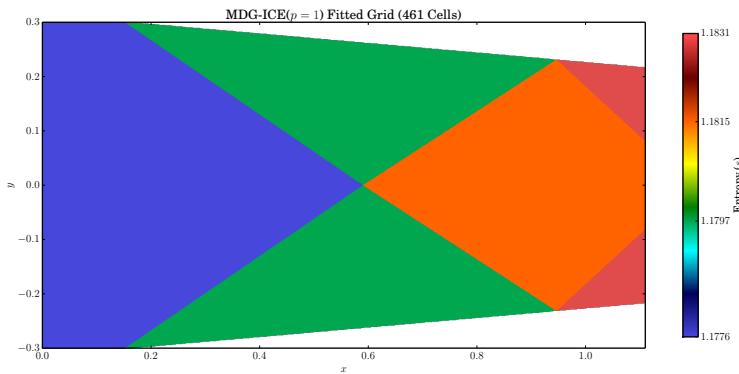
Fig. 7: Intersecting oblique shocks: Density



(a) The entropy field computed using $DG(p = 1)$ with shock capturing on a coarse grid. A significant amount of spurious entropy generation is visible.

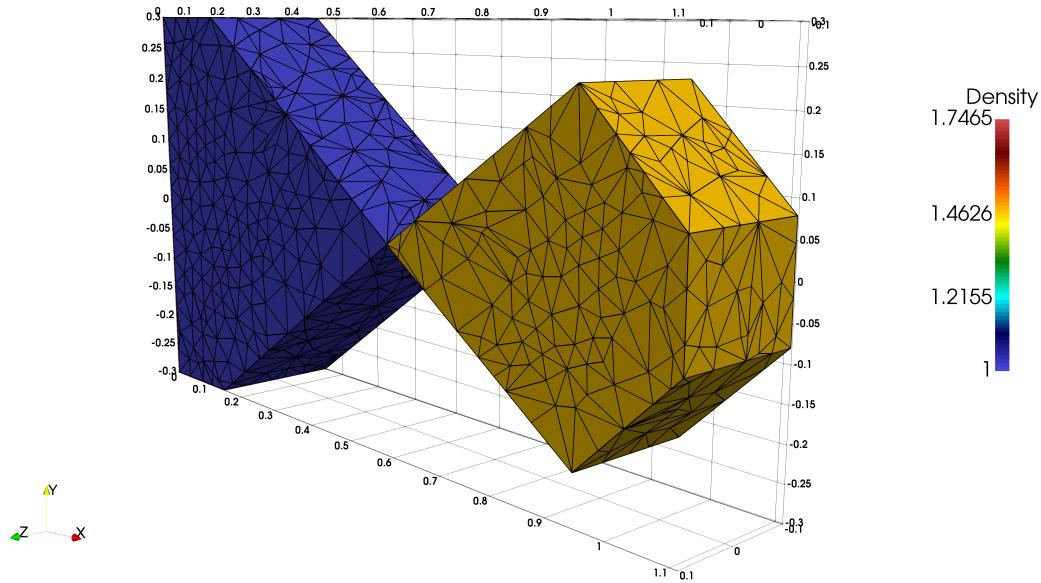


(b) The entropy field computed using $DG(p = 1)$ with shock capturing on the adapted grid. Spurious entropy generation has been reduced from (8a), however is still significant.

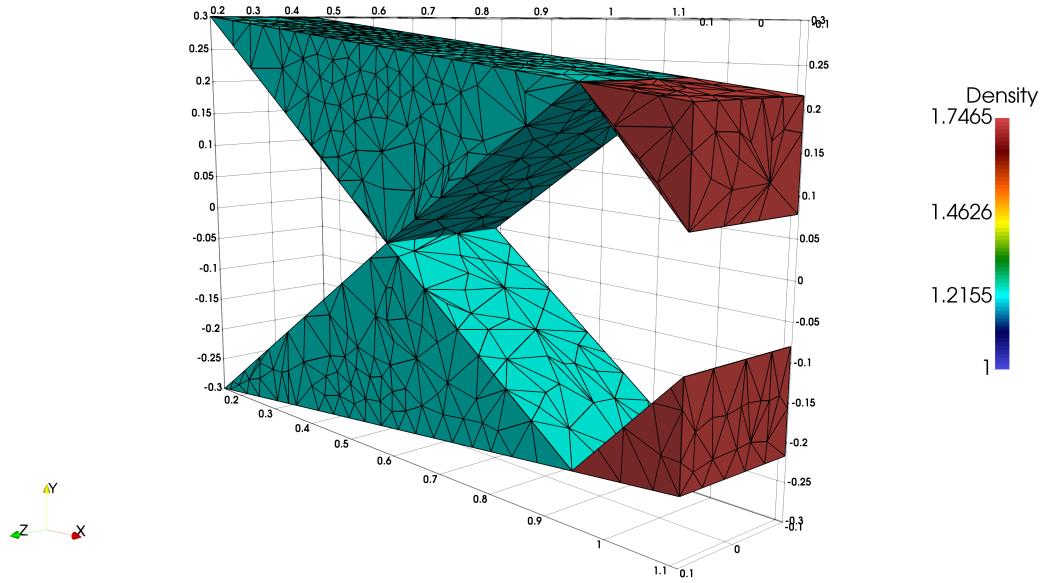


(c) The entropy field computed using $MDG-ICE(p = 1)$ without shock capturing.

Fig. 8: Intersecting Oblique Shocks: Entropy



(a) The density field computed using MDG-ICE($p = 1$) without shock capturing: thresholded to show ρ_1 and ρ_3 .



(b) The density field computed using MDG-ICE($p = 1$) without shock capturing: thresholded to show ρ_2 and ρ_4 .

Fig. 9: Three-Dimensional Intersecting Shocks

This case indicates that MDG-ICE generalizes to the case of curved interface geometry. Moreover, this result further illustrates the potential of MDG-ICE to circumvent the loss of accuracy that occurs using shock capturing.

6. CONCLUSIONS AND FUTURE WORK

The proposed method, MDG-ICE, accurately and stably computes flows with interfaces, including shocks, without relying on interface or shock capturing, within a unified formulation, by enforcing the conservation law and its interface condition simultaneously, while treating the discrete domain geometry as a variable. Preliminary test cases demonstrate that the method can be used to compute both steady and unsteady flows, a priori unknown interface topology, point singularities, using higher-order elements in arbitrary-dimensional spaces.

In order to further assess the robustness of the proposed method, future work will apply MDG-ICE to problems of increased complexity. For example, in Section 5.1, a slowly moving discontinuity was computed without artificial dissipation being introduced. Analogous results, which are oscillation-free, can be obtained using MDG-ICE for the case of slowly moving shocks in the context of the spacetime Euler equations, which has been reported as being challenging for Godunov-type or shock capturing methods, cf. [69–71]. Furthermore, in such situations, unfitted shocks can induce oscillations using standard methods, which shock capturing can effectively suppress. However, the challenge then becomes simultaneously preserving other low-amplitude flow structures such as physical acoustic waves, so that the shock capturing does not mistakenly suppress such waves too. This issue is of particular concern in aeroacoustics, and remains an open issue for higher-order methods [20]. This issue arises in other applications, including in scramjets, in which the flame can be quite sensitive to perturbations of the local flow conditions, or boundary layer separation on hypersonic vehicles. In the present work, we have demonstrated the ability of MDG-ICE to stably resolve shocked flows without artificial dissipation. In future work, we will assess MDG-ICE’s ability to resolve the interaction of smooth flow features with shocks.

Future work will further improve the robustness and efficiency of the method. Techniques such as domain decomposition (for spacetime marching) and multigrid are of particular interest to improve the linear solver efficiency. The initialization and regularization strategy used by the nonlinear solver will be optimized to achieve robust convergence using a minimum number of iterations. Future work will also consider the extension of the proposed formulation to flows involving diffusion and reactions, incompressibility constraints, as well as other conservation laws. While in the present work the formulation was introduced and preliminary comparisons demonstrated the accuracy and stability of the proposed method, a future theoretical study of convergence and stability will be used to either justify optimal high-order convergence in the presence of discontinuous interface or provide guidance to revise the method to ensure such properties.

Acknowledgements

This work was sponsored by the Office of Naval Research through the Naval Research Laboratory 6.1 Computational Physics Task Area. The authors acknowledge Devon Wood-Thomas for help implementing the grid refinement and edge collapse algorithms and Aaditya Singh for helpful discussions on linear solvers and preconditioners. In addition, the authors acknowledge Rainald Löhner, Johann Dahm, Brian Taylor, Ryan Johnson, Doug Schwer, Ravi Ramamurti, David Mott, and Gopal Patnaik for helpful discussions and providing feedback on this work.

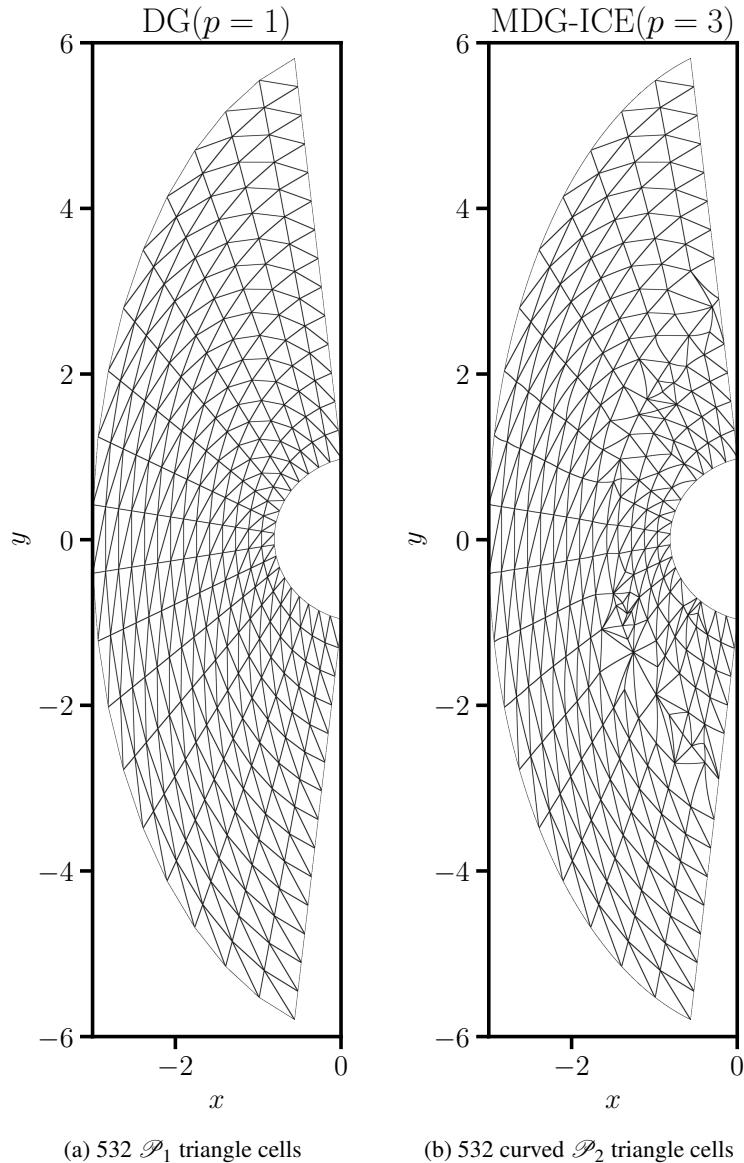


Fig. 10: The initial linear and quadratic grids, as well as the final shock fit grid, for the bow shock simulation. The $\text{DG}(p = 1)$ solver used shock capturing on a grid with 532 \mathcal{P}_1 triangle cells. The $\text{MDG-ICE}(p = 3)$ solver was initialized by projecting the $\text{DG}(p = 1)$ solution, and, in conjunction with degenerate cell collapse and local refinement, successfully fit the shock using curved \mathcal{P}_2 triangle cells, without shock capturing.

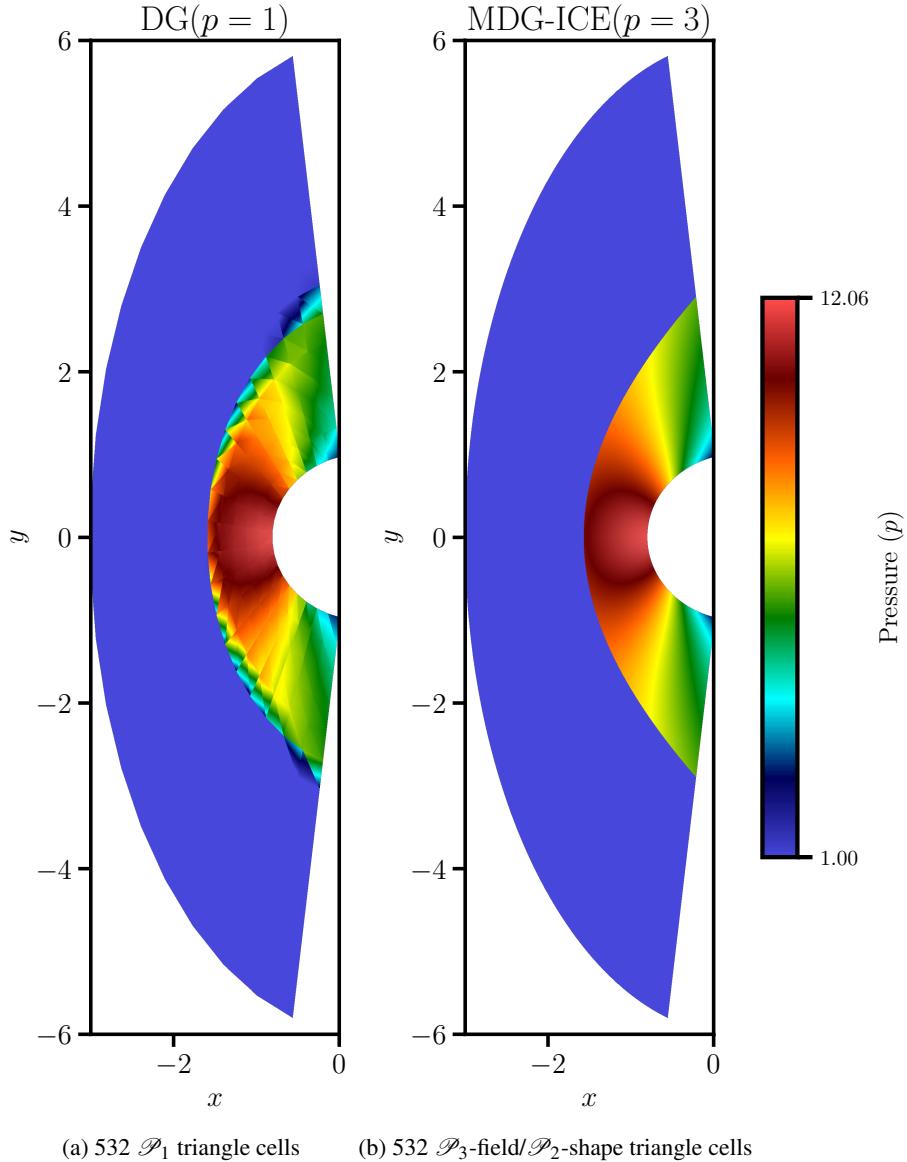


Fig. 11: The computed pressure field corresponding to the grids shown in Figure 10. The MDG-ICE($p = 3$) solver was initialized by projecting the DG($p = 1$) solution. The flow field variables were resolved using \mathcal{P}_3 elements, without shock capturing, while the shock interface was fit using curved \mathcal{P}_2 shape representation. The location of the shock along the line $y = 0$ was computed as $x = -1.698$ for a stand-off distance of 0.698.

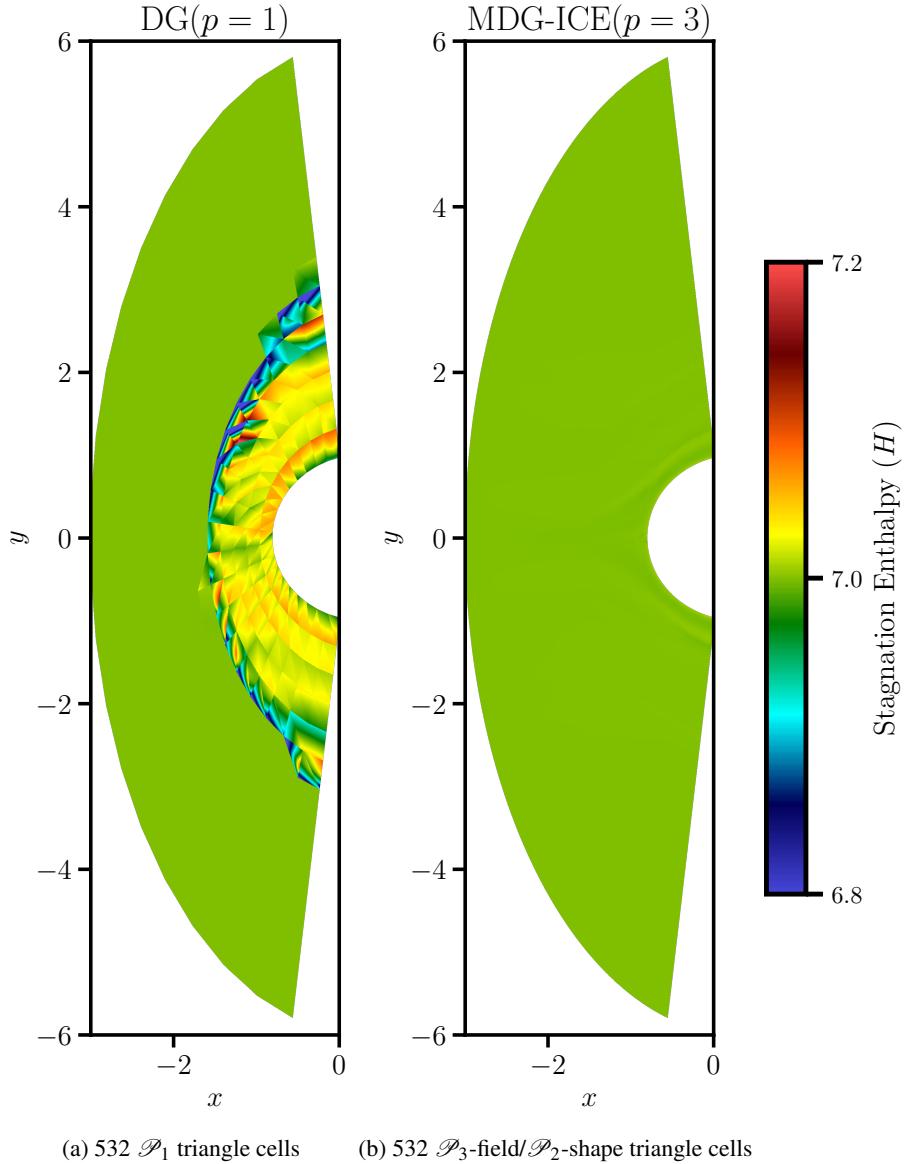
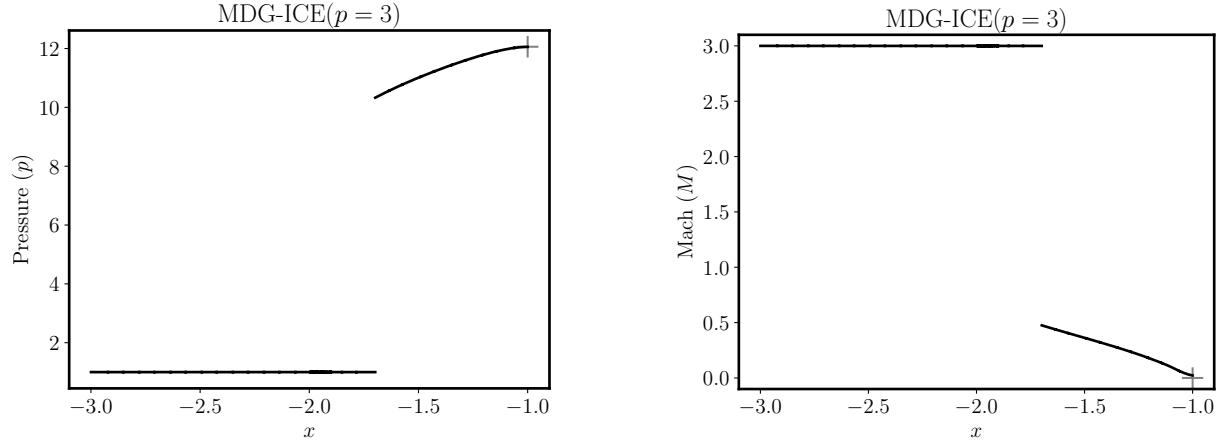


Fig. 12: The computed stagnation enthalpy (20) field corresponding to the grids shown in Figure 10. The exact solution has a constant stagnation enthalpy $H_\infty = 7$. The stagnation enthalpy field computed using MDG-ICE($p = 3$) without shock capturing, significantly improves the accuracy with respect to the exact solution, decreasing the error $\|H - H_\infty\|_{L_2(\Omega)}$ decreased from 1.2633×10^{-1} to 1.9704×10^{-3} .



(a) The pressure sampled along $y = 0$. The location of the shock was computed as $x = -1.698$ for a stand-off distance of 0.698. The exact pressure at the stagnation point, $p \approx 12.06096$, is marked with a cross. The computed pressure at the stagnation point on this coarse grid is 12.0559064.

(b) The Mach number sampled along $y = 0$. The exact Mach number at the stagnation point, $M = 0$, is marked with a cross. The computed Mach number at the stagnation point on this coarse grid is $2.26753700 \times 10^{-2}$.

Fig. 13: Bow shock: Computed with MDG-ICE($p = 3$) using a \mathcal{P}_2 shape representation.

A version of this work has been submitted for consideration for publication in the Journal of Computational Physics.

REFERENCES

1. F. Bassi and S. Rebay, “A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations,” *Journal of Computational Physics* **131**(2), 267–279 (1997).
2. F. Bassi and S. Rebay, “High-order accurate discontinuous finite element solution of the 2D Euler equations,” *Journal of Computational Physics* **138**(2), 251–285 (1997).
3. B. Cockburn and C.-W. Shu, “The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems,” *Journal of Computational Physics* **141**(2), 199–224 (1998).
4. B. Cockburn, G. Karniadakis, and C.-W. Shu, “The development of discontinuous Galerkin methods,” in *Discontinuous Galerkin Methods*, pp. 3–50 (Springer, 2000).
5. D. Arnold, F. Brezzi, B. Cockburn, and L. Marini, “Unified analysis of discontinuous Galerkin methods for elliptic problems,” *SIAM Journal on Numerical Analysis* **39**(5), 1749–1779 (2002).
6. R. Hartmann and P. Houston, “Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations,” *Journal of Computational Physics* **183**(2), 508–532 (2002).
7. K. Fidkowski, T. Oliver, J. Lu, and D. Darmofal, “p-Multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier–Stokes equations,” *Journal of Computational Physics* **207**(1), 92–113 (2005).

8. P.-O. Persson and J. Peraire, “Newton-GMRES preconditioning for discontinuous Galerkin discretizations of the Navier–Stokes equations,” *SIAM Journal on Scientific Computing* **30**(6), 2709–2733 (2008).
9. R. Hartmann and T. Leicht, “Higher order and adaptive DG methods for compressible flows,” in H. Deconinck, ed., *VKI LS 2014-03: 37th Advanced VKI CFD Lecture Series: Recent developments in higher order methods and industrial application in aeronautics, Dec. 9-12, 2013* (Von Karman Institute for Fluid Dynamics, Rhode Saint Genèse, Belgium, 2014).
10. J.-C. Lu, *An a posteriori error control framework for adaptive precision optimization using discontinuous Galerkin finite element method*, PhD thesis (Massachusetts Institute of Technology, 2005).
11. R. Hartmann, “Adjoint consistency analysis of discontinuous Galerkin discretizations,” *SIAM Journal on Numerical Analysis* **45**(6), 2671–2696 (2007).
12. R. Hartmann and T. Leicht, “Generalized adjoint consistent treatment of wall boundary conditions for compressible flows,” *Journal of Computational Physics* **300**, 754–778 (2015).
13. J. van der Vegt and H. Van der Ven, “Space–time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows: I. general formulation,” *Journal of Computational Physics* **182**(2), 546–585 (2002).
14. C. Klaij, J. van der Vegt, and H. van der Ven, “Space–time discontinuous Galerkin method for the compressible Navier–Stokes equations,” *Journal of Computational Physics* **217**(2), 589–611 (2006).
15. C. Lehrenfeld, “The Nitsche XFEM-DG space-time method and its implementation in three space dimensions,” *SIAM Journal on Scientific Computing* **37**(1), A245–A270 (2015).
16. R. Hartmann, “Adaptive discontinuous Galerkin methods with shock-capturing for the compressible Navier–Stokes equations,” *International Journal for Numerical Methods in Fluids* **51**(9–10), 1131–1156 (2006).
17. P.-O. Persson and J. Peraire, “Sub-cell shock capturing for discontinuous Galerkin methods,” *AIAA paper* **112**, 2006 (2006).
18. G. Barter and D. Darmofal, “Shock capturing with PDE-based artificial viscosity for DGFEM: Part I. formulation,” *Journal of Computational Physics* **229**(5), 1810–1827 (2010).
19. A. Huerta, E. Casoni, and J. Peraire, “A simple shock-capturing technique for high-order discontinuous Galerkin methods,” *International Journal for Numerical Methods in Fluids* **69**(10), 1614–1632 (2012).
20. Z. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. Huynh, N. Kroll, G. May, P.-O. Persson, B. van Leer, and M. Visbal, “High-order CFD methods: Current status and perspective,” *International Journal for Numerical Methods in Fluids* (2013), doi: 10.1002/fld.3767.
21. G. Moretti, “Thirty-six years of shock fitting,” *Computers & Fluids* **31**(4), 719–723 (2002).
22. M. Salas, *A shock-fitting primer* (CRC Press, 2009).
23. M. Salas, “A brief history of shock-fitting,” in *Computational Fluid Dynamics 2010*, pp. 37–53 (Springer, 2011).

24. R. Paciorri and A. Bonfiglioli, “A shock-fitting technique for 2D unstructured grids,” *Computers & Fluids* **38**(3), 715–726 (2009).
25. R. Paciorri and A. Bonfiglioli, “Shock interaction computations on unstructured, two-dimensional grids using a shock-fitting technique,” *Journal of Computational Physics* **230**(8), 3155–3177 (2011).
26. A. Bonfiglioli, M. Grottadaurea, R. Paciorri, and F. Sabetta, “An unstructured, three-dimensional, shock-fitting solver for hypersonic flows,” *Computers & Fluids* **73**, 162–174 (2013).
27. A. Bonfiglioli and R. Paciorri, “Convergence analysis of shock-capturing and shock-fitting solutions on unstructured grids,” *AIAA Journal* (2014).
28. A. Bonfiglioli, R. Paciorri, and L. Campoli, “Unsteady shock-fitting for unstructured grids,” *International Journal for Numerical Methods in Fluids* **81**(4), 245–261 (2016), ISSN 1097-0363, doi: 10.1002/fld.4183.
29. J. Glimm, J. Grove, X. Li, K.-M. Shyue, Y. Zeng, and Q. Zhang, “Three-dimensional front tracking,” *SIAM Journal on Scientific Computing* **19**(3), 703–727 (1998).
30. D. She, R. Kaufman, H. Lim, J. Melvin, A. Hsu, and J. Glimm, “Front-tracking methods,” *Handbook of Numerical Analysis* **17**, 383–402 (2016).
31. V.-T. Nguyen, J. Peraire, B. Khoo, and P.-O. Persson, “A discontinuous Galerkin front tracking method for two-phase flows with surface tension,” *Computers & Fluids* **39**(1), 1–14 (2010).
32. T.-P. Fries and T. Belytschko, “The extended/generalized finite element method: an overview of the method and its applications,” *International Journal for Numerical Methods in Engineering* **84**(3), 253–304 (2010).
33. A. Hansbo and P. Hansbo, “An unfitted finite element method, based on Nitsche’s method, for elliptic interface problems,” *Computer Methods in Applied Mechanics and Engineering* **191**(47), 5537–5552 (2002).
34. S. Groß, V. Reichelt, and A. Reusken, “A finite element based level set method for two-phase incompressible flows,” *Computing and Visualization in Science* **9**(4), 239–257 (2006).
35. R. Becker, E. Burman, and P. Hansbo, “A Nitsche extended finite element method for incompressible elasticity with discontinuous modulus of elasticity,” *Computer Methods in Applied Mechanics and Engineering* **198**(41), 3352–3360 (2009).
36. R. Massjung, “An unfitted discontinuous Galerkin method applied to elliptic interface problems,” *SIAM Journal on Numerical Analysis* **50**(6), 3134–3162 (2012).
37. C. Lehrenfeld and A. Reusken, “Analysis of a Nitsche xfem-dg discretization for a class of two-phase mass transport problems,” *SIAM Journal on Numerical Analysis* **51**(2), 958–983 (2013).
38. C. Lehrenfeld, “High order unfitted finite element methods on level set domains using isoparametric mappings,” *Computer Methods in Applied Mechanics and Engineering* **300**, 716–733 (2016).
39. V. Dobrev, T. Ellis, T. Kolev, and R. Rieben, “Curvilinear finite elements for Lagrangian hydrodynamics,” *International Journal for Numerical Methods in Fluids* **65**(11-12), 1295–1310 (2011).

40. V. Dobrev, T. Kolev, and R. N. Rieben, “High-order curvilinear finite element methods for Lagrangian hydrodynamics,” *SIAM Journal on Scientific Computing* **34**(5), B606–B641 (2012).
41. R. Anderson, V. Dobrev, T. Kolev, and R. Rieben, “Monotonicity in high-order curvilinear finite element arbitrary Lagrangian–Eulerian remap,” *International Journal for Numerical Methods in Fluids* **77**(5), 249–273 (2015).
42. P.-O. Persson, J. Bonet, and J. Peraire, “Discontinuous Galerkin solution of the Navier–Stokes equations on deformable domains,” *Computer Methods in Applied Mechanics and Engineering* **198**(17), 1585–1595 (2009).
43. K. Fidkowski, “A hybridized discontinuous Galerkin method on mapped deforming domains,” *Computers & Fluids* **139**, 80–91 (2016).
44. M. Neumüller, *Space-Time Methods: Fast Solvers and Applications*, PhD thesis (Graz University of Technology, June 2013).
45. A. Harten and J. Hyman, “Self adjusting grid methods for one-dimensional hyperbolic conservation laws,” *Journal of Computational Physics* **50**(2), 235–269 (1983).
46. K. Miller and R. Miller, “Moving finite elements. I,” *SIAM Journal on Numerical Analysis* **18**(6), 1019–1032 (1981).
47. K. Miller, “Moving finite elements. II,” *SIAM Journal on Numerical Analysis* **18**(6), 1033–1057 (1981).
48. R. Gelinas, S. Doss, and K. Miller, “The moving finite element method: applications to general partial differential equations with multiple large gradients,” *Journal of Computational Physics* **40**(1), 202–249 (1981).
49. P. Roe and H. Nishikawa, “Adaptive grid generation by minimizing residuals,” *International Journal for Numerical Methods in Fluids* **40**(1-2), 121–136 (2002).
50. D. Sanjaya and K. Fidkowski, “Improving high-order finite element approximation through geometrical warping,” *AIAA Journal* **54**(12), 3994–4010 (2016).
51. M. Giles and N. Pierce, “An introduction to the adjoint approach to design,” *Flow, Turbulence, and Combustion* **65**, 393–415 (2000).
52. M. Giles, “Discrete adjoint approximations with shocks,” in *Hyperbolic Problems: Theory, Numerics, Applications*, pp. 185–194 (Springer, 2003).
53. M. Giles and S. Ulbrich, “Convergence of linearized and adjoint approximations for discontinuous solutions of conservation laws. part 1: Linearized approximations and linearized output functionals,” *SIAM Journal on Numerical Analysis* **48**(3), 882–904 (2010).
54. M. Giles and S. Ulbrich, “Convergence of linearized and adjoint approximations for discontinuous solutions of conservation laws. part 2: Adjoint approximations and extensions,” *SIAM Journal on Numerical Analysis* **48**(3), 905–921 (2010).
55. N. Nguyen, J. Peraire, and B. Cockburn, “An implicit high-order hybridizable discontinuous Galerkin method for linear convection–diffusion equations,” *Journal of Computational Physics* **228**(9), 3232–3254 (2009).

56. J. Peraire, N. Nguyen, and B. Cockburn, "A hybridizable discontinuous Galerkin method for the compressible Euler and Navier-Stokes equations," *AIAA paper* **363**, 2010 (2010).
57. M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich, *Optimization with PDE Constraints* (Springer, 2009).
58. C. Paige and M. Saunders, "Solution of sparse indefinite systems of linear equations," *SIAM Journal on Numerical Analysis* **12**(4), 617–629 (1975).
59. C. Paige and M. Saunders, "LSQR: An algorithm for sparse linear equations and sparse least squares," *ACM Transactions on Mathematical Software* **8**(1), 43–71 (1982).
60. D. Fong and M. Saunders, "LSMR: An iterative algorithm for sparse least-squares problems," *SIAM Journal on Scientific Computing* **33**(5), 2950–2971 (2011).
61. N. Gould and J. Scott, "The state-of-the-art of preconditioners for sparse linear least-squares problems," *ACM Transactions on Mathematical Software (TOMS)* **43**(4), 36 (2017).
62. R. Löhner, *Applied CFD Techniques* (J. Wiley & Sons, 2008).
63. M. Feistauer, J. Felcman, and I. Straškraba, *Mathematical and computational methods for compressible flow* (Oxford University Press, 2003).
64. E. Toro, *Riemann solvers and numerical methods for fluid dynamics: a practical introduction* (Springer Science & Business Media, 2013).
65. P. Batten, N. Clarke, C. Lambert, and D. Causon, "On the choice of wavespeeds for the hllc riemann solver," *SIAM Journal on Scientific Computing* **18**(6), 1553–1570 (1997).
66. C. Shu, "Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws," in *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, pp. 325–432 (Springer, 1998).
67. J. Quirk, "A contribution to the great Riemann solver debate," *International Journal for Numerical Methods in Fluids* **18**(6), 555–574 (1994).
68. M. Pandolfi and D. D'Ambrosio, "Numerical instabilities in upwind methods: analysis and cures for the "carbuncle" phenomenon," *Journal of Computational Physics* **166**(2), 271–301 (2001).
69. P. Colella and P. Woodward, "The piecewise parabolic method (PPM) for gas-dynamical simulations," *Journal of Computational Physics* **54**(1), 174–201 (1984).
70. M. Arora and P. Roe, "On postshock oscillations due to shock capturing schemes in unsteady flows," *Journal of Computational Physics* **130**(1), 25–40 (1997).
71. E. Johnsen and S. Lele, "Numerical errors generated in simulations of slowly moving shocks," *Center for Turbulence Research, Annual Research Briefs* pp. 1–12 (2008).